– EPE – NEW NAME NEW DESIGN!

Practical Electronics

The UK's premier electronics and computing maker magazine

Audio Out
Super low-noise power supply for your theremin

Practically Speaking
Getting to grips with surface-mount ICs

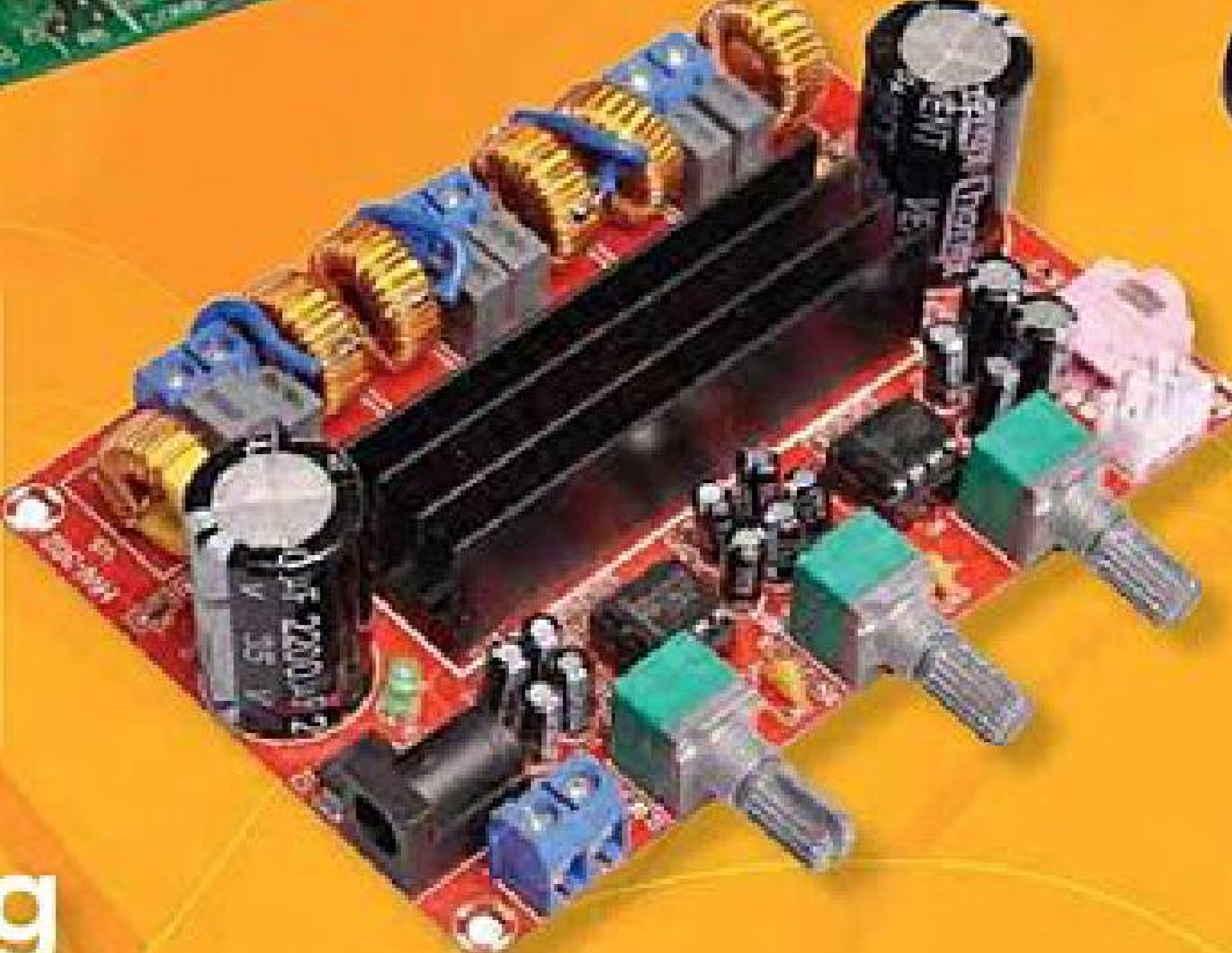
Circuit Surgery
LTspice – Using behavioural sources

Micromite LCD BackPack V3

WIN!
Microchip J-32 Debug Probe
WIN!

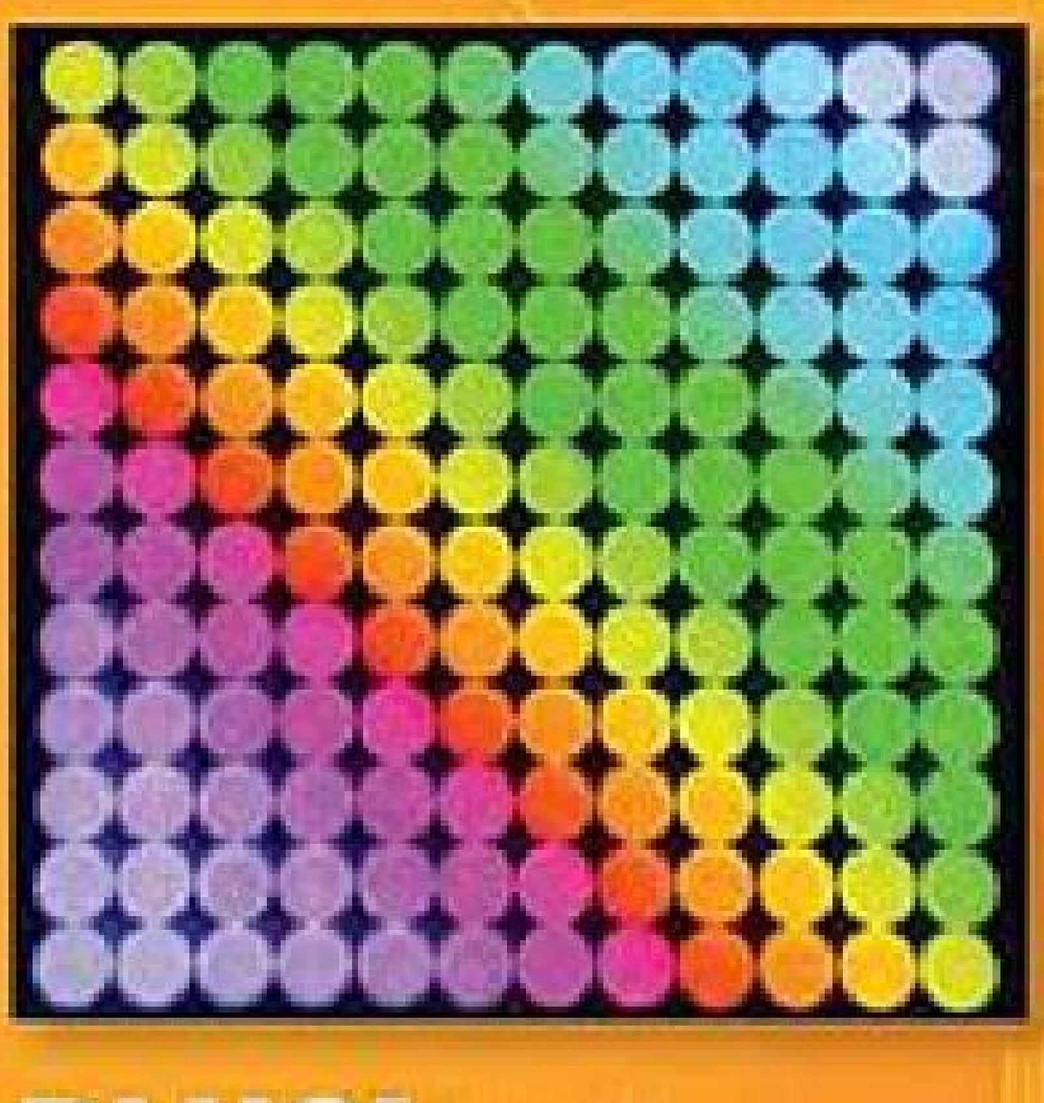
Don't miss our Super Summer Sale!

Steering Wheel Audio Button Adaptor

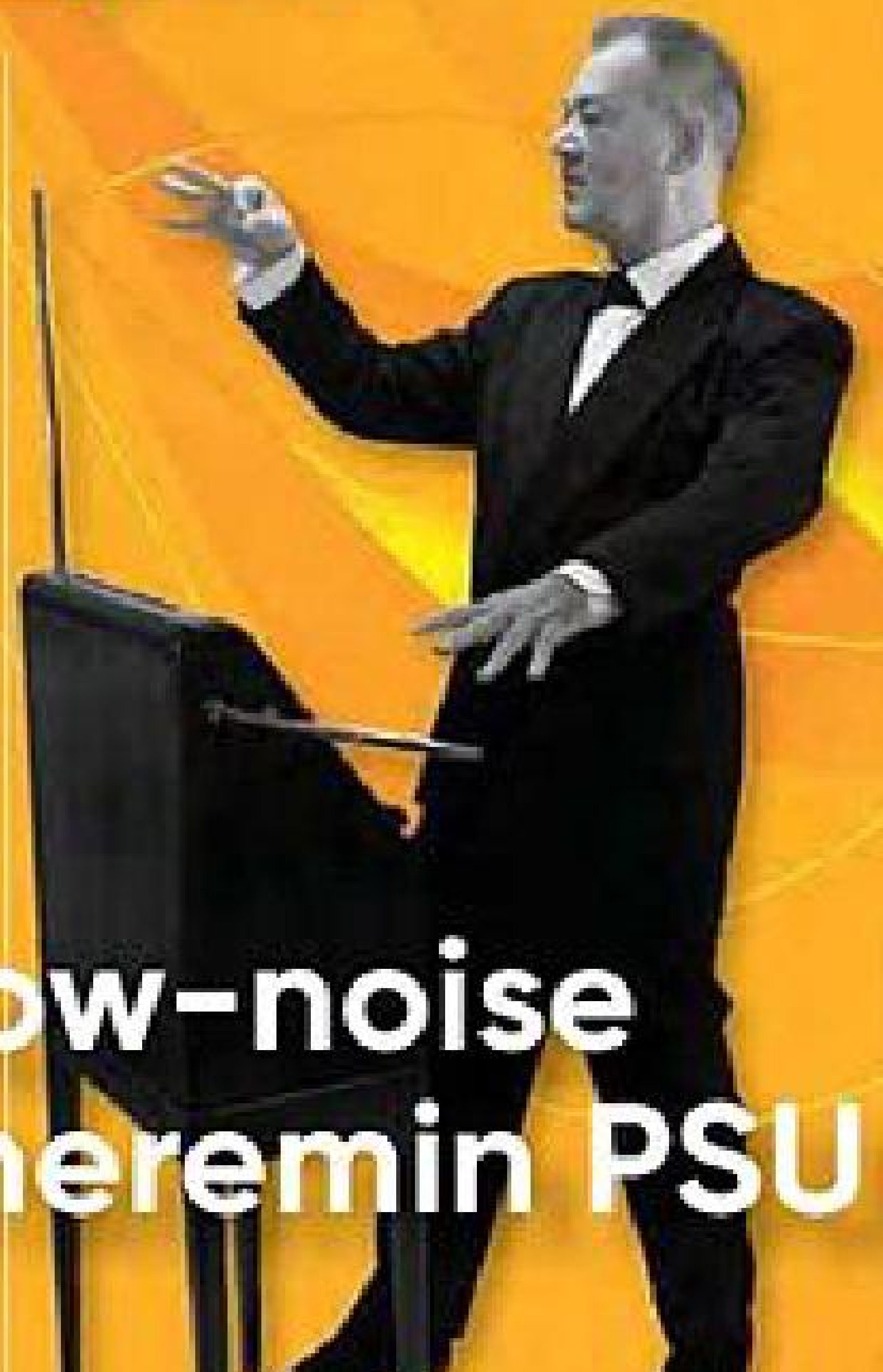


Bargain Class-D Amplifier

Ping-pong ball lighting!



IR control of Robot Buggy



Low-noise theremin PSU

PLUS!
Net Work – Yubico's latest Security Key
Techno Talk – The benefits of hindsight
Electronic Building Blocks – Battery capacity tester

Aug 2020 £4.99
9 772632 573016 08

www.electronpublishing.com | @practicalelec | practicalelectronics

Introducing a New Generation of AVR® Devices

AVR-DA Flexibility in Design

The Functional Safety Ready AVR-DA family is the new generation of AVR microcontrollers running up to 24 MHz across the full supply voltage range 1.8V to 5.5V featuring intelligent analog and core independent peripherals.

The AVR-DA family uses the latest Core Independent Peripherals and low-power capability to handle real-time control functions in industrial control, home appliance, automotive and Internet of Things (IoT) applications. The Event System and Configurable Custom Logic (CCL) peripherals offer near-zero latency for time-critical functions. Intelligent Analog peripherals like 12-bit differential Analog-to-Digital Converter (ADC), Zero-Cross Detect (ZCD), 10-bit Digital-to-Analog Converter (DAC) and the latest-generation Peripheral Touch Controller (PTC) put the AVR-DA family in the center of sensor interface, system monitoring and human interface applications. 5V operation for increased noise immunity and low-power features ease power supply concerns.

- Product certifications made easier with Functional Safety supporting documentation and development tools.
- Add IoT and connectivity to your application, thanks to the high memory densities and SRAM-to-Flash ratio.
- Design robust and noise-immune applications with built-in safety features and 5V supply.
- Create state-of-the-art capacitive touch interfaces using the latest-generation PTC.

MICROCHIP
AVR128DA64

Functional Safety Ready

www.microchip.com/avr-da

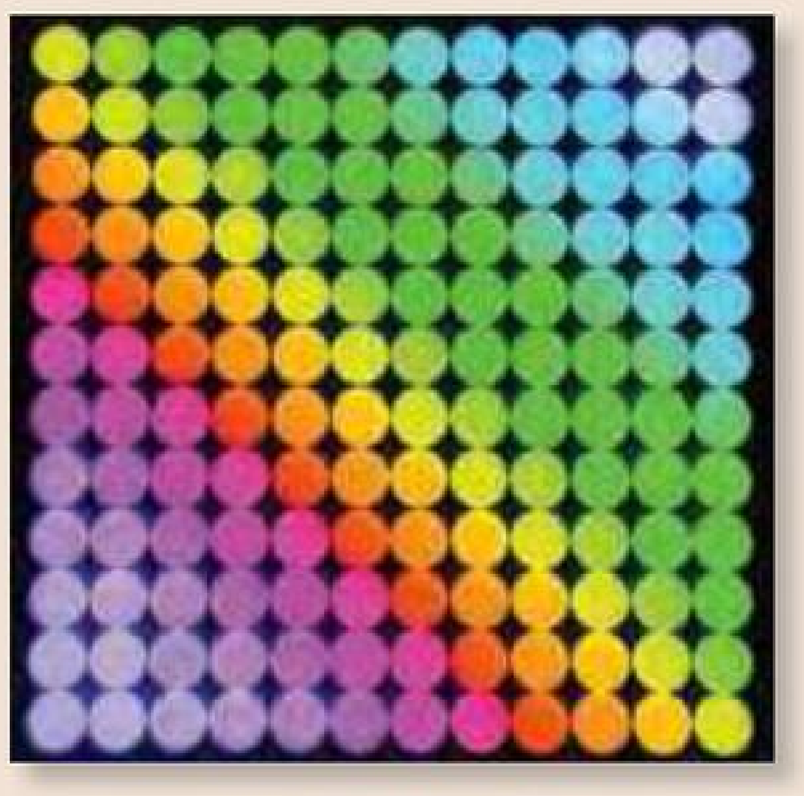
MICROCHIP

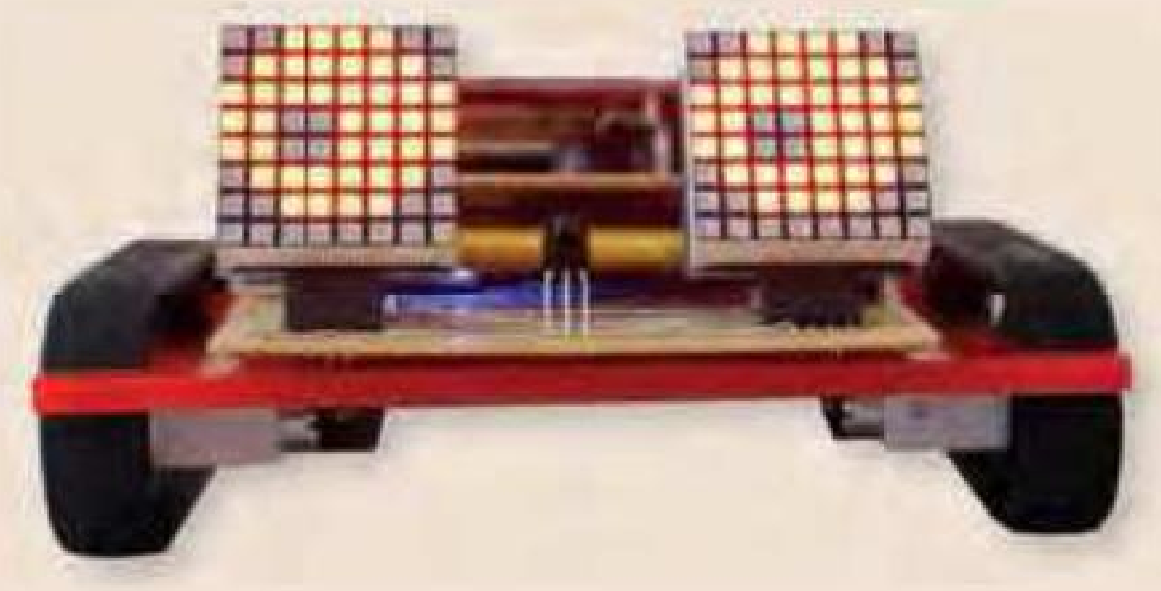
The Microchip name and logo, the Microchip logo and AVR are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
All other trademarks are the property of their registered owners.
© 2020 Microchip Technology Inc. All rights reserved. DS30010220A. MEC2321A-ENG-05-20

Practical Electronics

Volume 49. No. 8
August 2020
ISSN 2632 573X

Contents





Made in the UK.

Written in Britain, Australia, the US and Ireland.

Read everywhere.

© Electron Publishing Limited 2020

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The September 2020 issue of *Practical Electronics* will be published on Thursday, 6 August 2020 – see page 72.

Projects and Circuits

Micromite LCD BackPack V3 by Tim Blythman 16
All the features of the V1 and V2 BackPacks, plus it supports both 2.8-inch and 3.5-inch touchscreen displays and boasts five new optional features.

Steering Wheel Audio Button to Infrared Adaptor by John Clarke 25
This adaptor lets you use most push-button steering wheel controls with a wide range of aftermarket head units.

Junk Mail Repeller by Allan Linton-Smith 32
Is your letterbox full of junk, even though you have a NO JUNK MAIL sign? If so, you need to build our Junk Mail Repeller!

Class-D Stereo Plus Subwoofer Amplifier by Allan Linton-Smith 38
If we told you that you could get an assembled 3 × 50W amplifier module for £7.50, you'd probably think it's junk. But in this circuit, that isn't the case!

Series, Features and Columns

The Fox Report by Barry Fox 8
New tech norm

Techno Talk by Mark Nelson 10
The benefits of hindsight

Net Work by Alan Winstanley 12
In barely a dozen weeks we've entered a different world. This month, Net Work looks at the accelerated rise of video conference calls and Yubico's latest Security Key.

Circuit Surgery by Ian Bell 43
LTspice – Using behavioural sources

Audio Out by Jake Rothman 48
Low-noise Theremin Power Supply – Part 1

Make it with Micromite by Phil Boyce 52
Part 19: Controlling your MRB with an IR transmitter and animating its eyes

Practically Speaking by Mike Hibbett 56
Introduction to surface-mount technology – Part 3

Max's Cool Beans by Max The Magnificent 59
Flashing LEDs and drooling engineers – Part 6

Electronic Building Blocks by Julian Edgar 64
Battery capacity tester

Regulars and Services

Wireless for the Warrior 2
PE Summer Sale! 3
Subscribe to Practical Electronics and save money 4
NEW! Practical Electronics back issues DOWNLOADS – great 20-year deal! 6
Reader services – Editorial and Advertising Departments 7
Editorial 7
At last... back issue downloads!
PE Teach-In 8 11
PE Teach-In 9 14
Exclusive Microchip reader offer 15
Win a Microchip J-32 Debug Probe
Practical Electronics – get your back issues here! 37
Direct Book Service 66
Build your library of carefully chosen technical books
Practical Electronics PCB Service 68
PCBs for Practical Electronics projects
Teach-In bundle – what a bargain! 70
Classified ads and Advertiser index 71
Next month! – highlights of our next issue of Practical Electronics 72

WIRELESS FOR THE WARRIOR

by LOUIS MEULSTEE

THE DEFINITIVE TECHNICAL HISTORY OF RADIO COMMUNICATION EQUIPMENT IN THE BRITISH ARMY

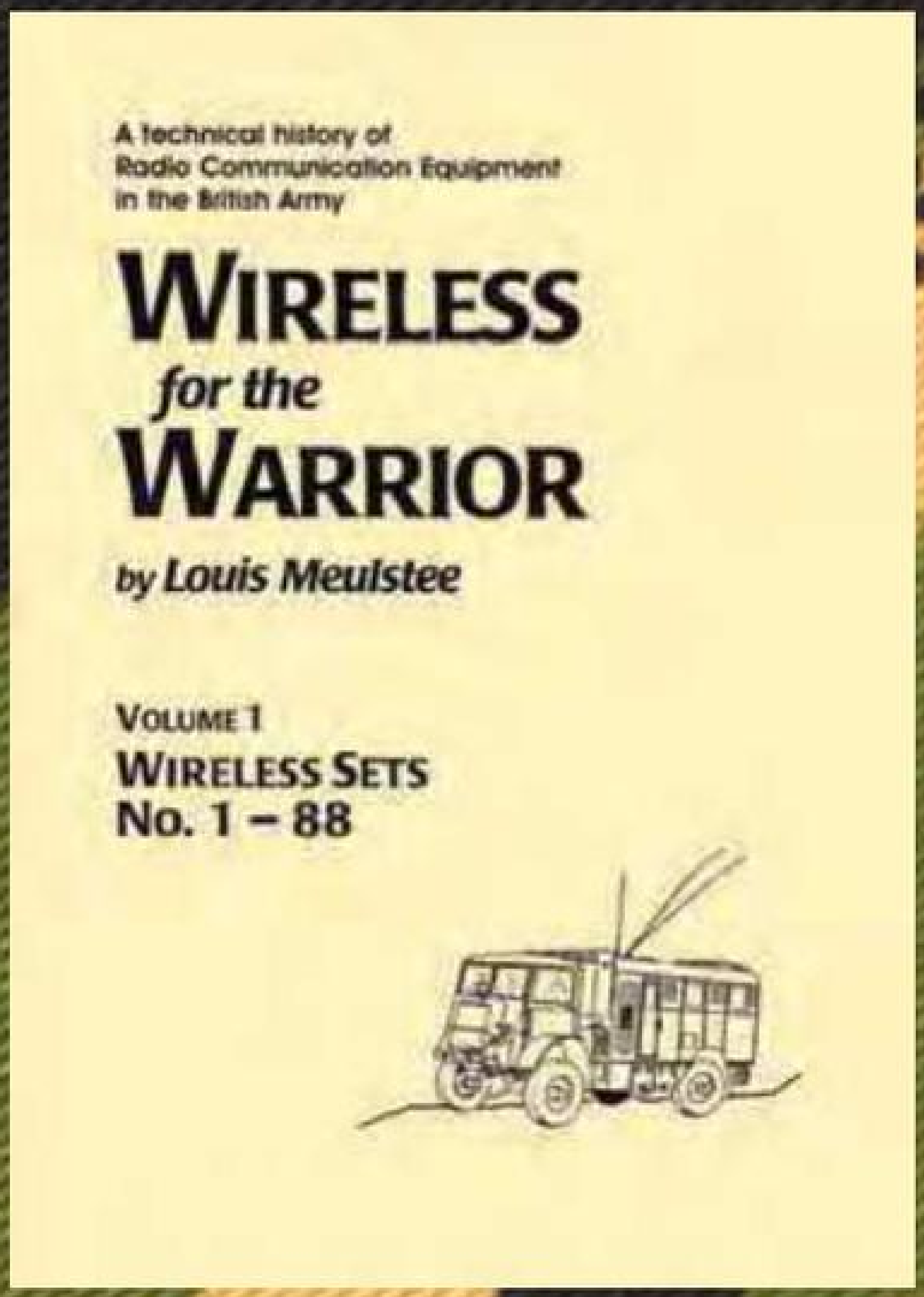
The *Wireless for the Warrior* books are a source of reference for the history and development of radio communication equipment used by the British Army from the very early days of wireless up to the 1960s.

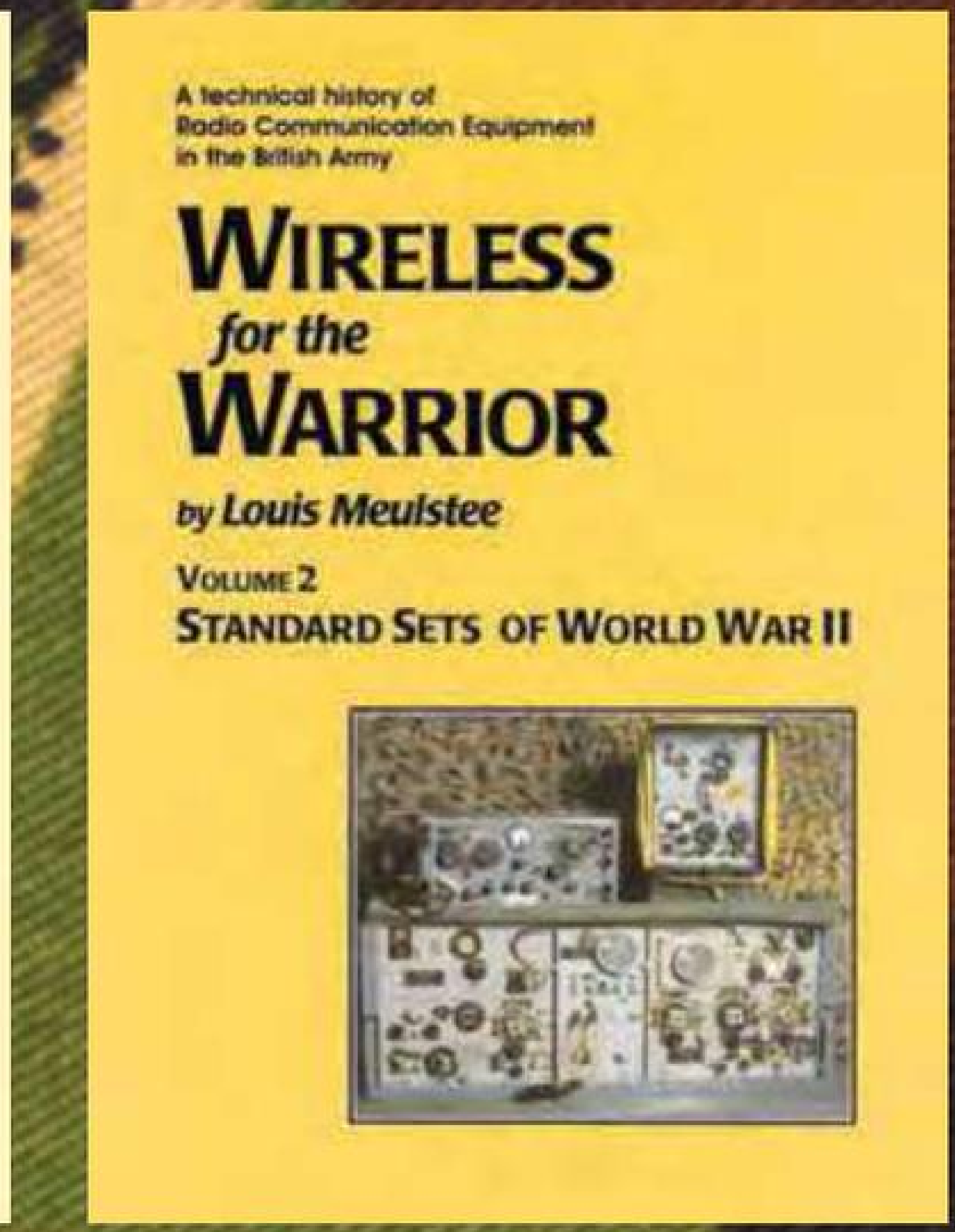
The books are very detailed and include circuit diagrams, technical specifications and alignment data, technical development history, complete station lists and vehicle fitting instructions.

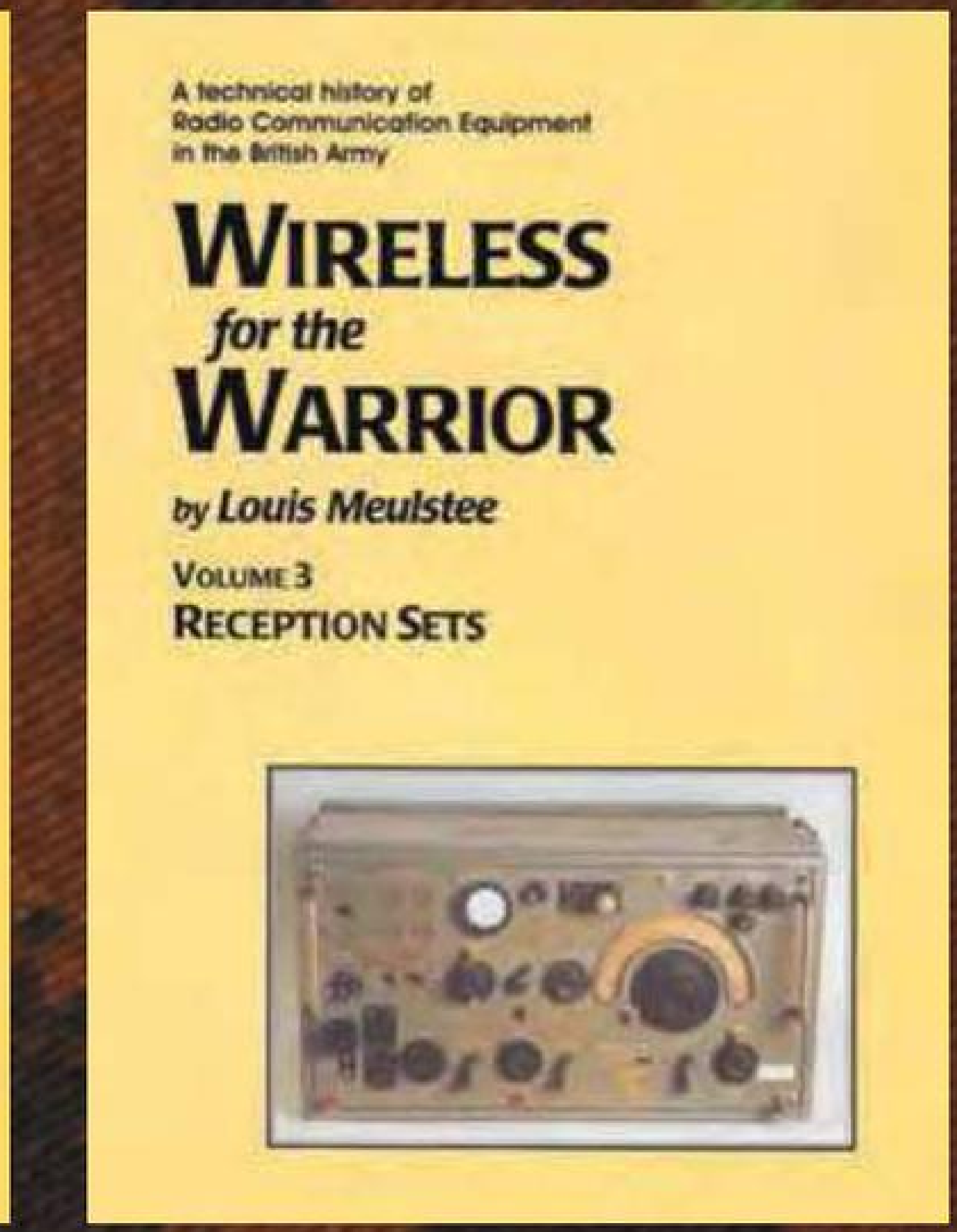
Volume 1 and *Volume 2* cover transmitters and transceivers used between 1932-1948. An era that starts with positive steps taken to formulate and develop a new series of wireless sets that offered great improvements over obsolete World War I pattern equipment. The other end of this timeframe saw the introduction of VHF FM and hermetically sealed equipment.

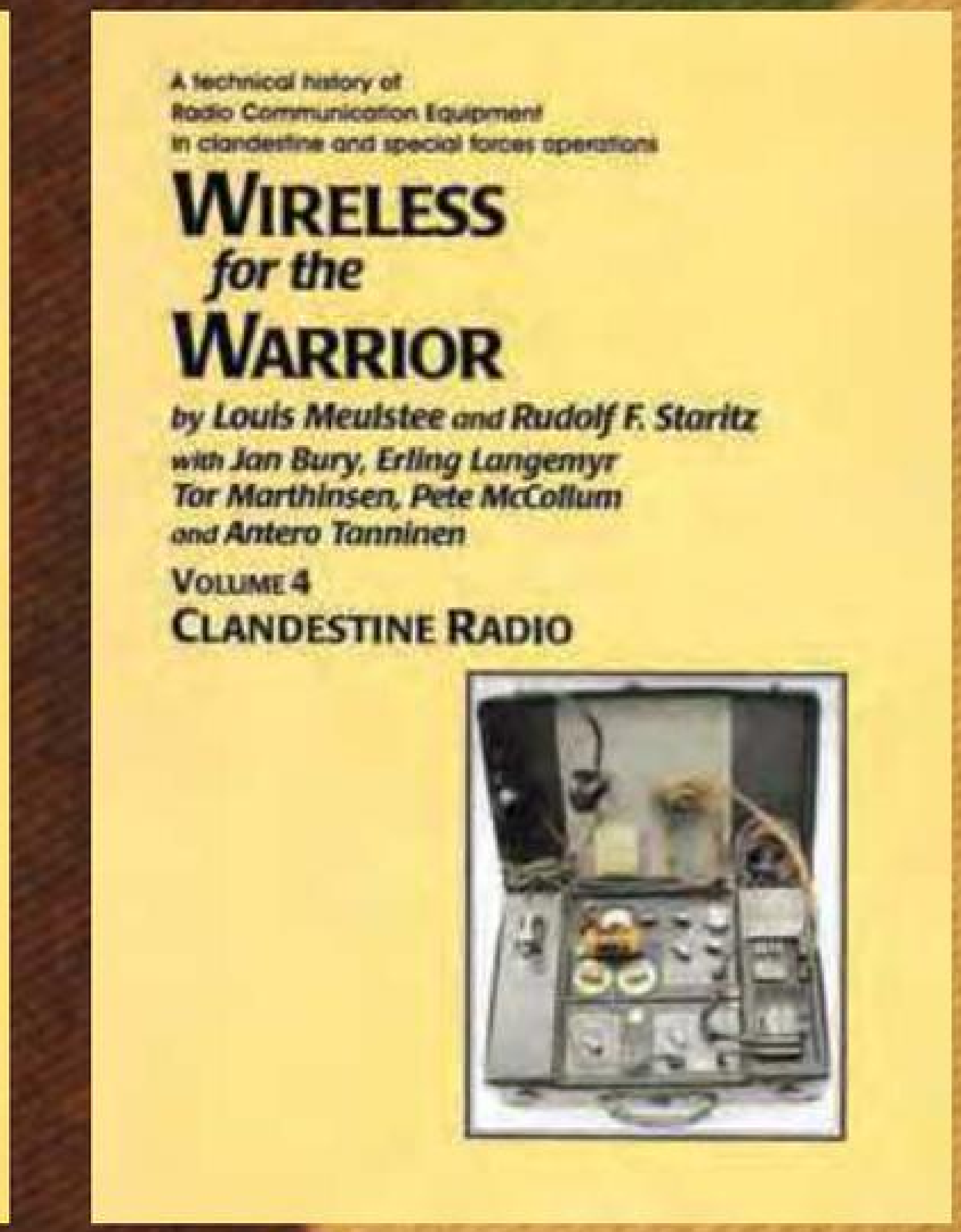
Volume 3 covers army receivers from 1932 to the late 1960s. The book not only describes receivers specifically designed for the British Army, but also the Royal Navy and RAF. Also covered: special receivers, direction finding receivers, Canadian and Australian Army receivers, commercial receivers adopted by the Army, and Army Welfare broadcast receivers.

Volume 4 covers clandestine, agent or 'spy' radio equipment, sets which were used by special forces, partisans, resistance, 'stay behind' organisations, Australian Coast Watchers and the diplomatic service. Plus, selected associated power sources, RDF and intercept receivers, bugs and radar beacons.


A technical history of Radio Communication Equipment in the British Army
WIRELESS for the WARRIOR
by Louis Meulstee
VOLUME 1
WIRELESS SETS No. 1 – 88


A technical history of Radio Communication Equipment in the British Army
WIRELESS for the WARRIOR
by Louis Meulstee
VOLUME 2
STANDARD SETS OF WORLD WAR II


A technical history of Radio Communication Equipment in the British Army
WIRELESS for the WARRIOR
by Louis Meulstee
VOLUME 3
RECEPTION SETS


A technical history of Radio Communication Equipment in clandestine and special forces operations
WIRELESS for the WARRIOR
by Louis Meulstee and Rudolf F. Staritz
with Jan Bury, Erling Langemyr
Tor Marthinsen, Pete McCollum
and Antero Tanninen
VOLUME 4
CLANDESTINE RADIO

ORDER YOURS TODAY!
JUST CALL 01202 880299 OR VISIT www.electronpublishing.com

Super Summer Sale!

Subscribe now and avoid future price rises!

Practical Electronics special offers

33% off all PCBs up to and including those in the December 2019 issue
25% off all PE/EPE print back issues up to the December 2019 issue

Practical Electronics subscription offer

Subscribe to PE hard copy for 12-months and receive a **free 12-month back-issue download** of your choice from Volume 28-46 (see page 6).

Subscribe to PE hard copy for 24-months and receive **two free 12-month back-issue downloads** of your choice from Volume 28-46 (see page 6), normal price.

Existing subscribers are welcome to renew early for another 12 or 24 months and still receive the offer – your new subscription will simply be added to the end of your existing subscription – **no issues lost by renewing early and you of course avoid future subscription price rises!**

Subscription offers only apply to direct subscriptions via Select Publishing Services, our distributor – not via third-party subscription services.

Payment must be received by close of offer: 30 September 2020

Don't miss out!
Just call 01202 880299 or visit our secure online shop at:
www.electronpublishing.com

– EPE –
NEW NAME
NEW DESIGN!

Practical Electronics

UK readers SAVE £1 on every issue

NEW subscriptions hotline!

01202 087631

Take out a one-year subscription and save £10 over the year.
Even better – save £1 per issue if you subscribe for two years – a total saving of £24.

Overseas rates represent exceptional value

You also:
- **Avoid any cover price increase for the duration of your subscription**
- **Get your magazine delivered to your door each month**
- **Guarantee a copy, even if the newsagents sell out**

Order by:
- **Phone or post with a cheque, postal order or credit card**
- **Or order online at: www.electronpublishing.com**

Practical Electronics

SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

UK: 6-months £26.99, 12-months £49.85, 24-months £94.99

Europe Airmail: 6-months £30.99, 12-months £57.99, 24-months £109.99

Rest Of The World Airmail: 6-months £37.99, 12-months £70.99, 24-months £135.99

Cheques (in **£ sterling only**) payable to *Practical Electronics* and sent to: Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

Also via our online shop at**: www.electronpublishing.com**

Subscriptions start with the next available issue. We accept MasterCard or Visa.
(For past issues see the Back Issues page.)

MasterCard VISA

DIGITAL SUBSCRIPTIONS

Electronic subscriptions are available from £24.99 for 12 months, more details at: **www.electronpublishing.com**

SUBSCRIPTION ORDER FORM

☐ **6 Months: UK £26.99, Europe £30.99 (Airmail), Rest of the World £37.99 (Airmail)**
☐ **1 Year: UK £49.85, Europe £57.99 (Airmail), Rest of the World £70.99 (Airmail)**
☐ **2 Years: UK £94.99, Europe £109.99 (Airmail), Rest of the World £135.99 (Airmail)**

To:
Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom

Tel: 01202 087631
Email: pesubs@selectps.com

I enclose payment of £ ....
(cheque/PO in £ sterling only)
payable to *Practical Electronics*

MasterCard VISA

☐ Please charge my Visa/Mastercard

My card number is: ....
Please print clearly, and check that you have the number correct

Card Security Code .... Valid From Date.......................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ....

Name ....

Address ....

Post code .... Tel. ....

Email ....

Subscriptions start with the next available issue.

STAY AT HOME!

AND READ

SILICON CHIP!

Of course we want you to continue to support Practical Electronics . . . but did you know you can also subscribe to SILICON CHIP – Australia's electronics magazine – direct from the United Kingdom (or anywhere else!)

And now there's a real choice:

(1) **A 12 month printed edition subscription** (12 issues), delivered right to your mailbox, for $AU150 per year (that's about £77 / €87 at time of publication)

(2) **A 12 month digital/online edition** (exactly the same content as the printed magazine) for $AU85 (about £44/ €50 at press time) – available to view when printed edition is released in Australia.

(3) **A 12 month combined printed and online subscription** – the best of both worlds – read now and have your printed edition available for future reference! That's just $AU170 for 12 months (about £87/ €98).

(4) **You can order a SILICON CHIP back issue** for $AU10 (about £5 / €6) plus P&P – most recent months still in stock

(5) **Now you can order SILICON CHIP archives on flash drive**: a quality metal flash drive containing any five year block of SILICON CHIP – from the first issue back in 1987 until Dec 2019 (see siliconchip.com.au/shop/digital_pdfs for details)

Log onto www.siliconchip.com.au for much more information!

You might also be interested in:

Radio, TV & Hobbies on DVD

Take a trip back in time for the entire Radio, TV and Hobbies magazine, from April 1939 through to March 1965 – ready to enjoy at your leisure, again and again and again.
Comes in a protective case – and it's just $AU62 plus p&p (about £31.60/€36)
Or you can download the digital edition of Radio, TV & Hobbies: $AU50 (£25.50 / €29)
See siliconchip.com.au/shop/3 for other options. And yes, you can order online!

SILICON CHIP

Australia's Electronics Magazine

DOWNLOAD YOUR PE/EPE BACK ISSUES

COMPLETE-YEAR collections – all 12 issues for each year!

From just £6.95 each

VOL 28	January to December 1999	**£6.95**
VOL 29	January to December 2000	**£6.95**
VOL 30	January to December 2001	**£6.95**
VOL 31	January to December 2002	**£6.95**
VOL 32	January to December 2003	**£6.95**
VOL 33	January to December 2004	**£6.95**
VOL 34	January to December 2005	**£6.95**
VOL 35	January to December 2006	**£6.95**
VOL 36	January to December 2007	**£6.95**
VOL 37	January to December 2008	**£6.95**
VOL 38	January to December 2009	**£8.95**
VOL 39	January to December 2010	**£8.95**
VOL 40	January to December 2011	**£8.95**
VOL 41	January to December 2012	**£12.95**
VOL 42	January to December 2013	**£12.95**
VOL 43	January to December 2014	**£14.95**
VOL 44	January to December 2015	**£14.95**
VOL 45	January to December 2016	**£16.95**
VOL 46	January to December 2017	**£18.95**
VOL 47	January to December 2018	**£22.95**
VOL 48	January to December 2019	**£24.95**

FIVE-YEAR collections – 60 issues!

From just £14.95 each

PE2832:	**Jan 99 to Dec 03**	**£14.95**
PE2933:	**Jan 00 to Dec 04**	**£14.95**
PE3034:	**Jan 01 to Dec 05**	**£14.95**
PE3135:	**Jan 02 to Dec 06**	**£14.95**
PE3236:	**Jan 03 to Dec 07**	**£14.95**
PE3337:	**Jan 04 to Dec 08**	**£14.95**
PE3438:	**Jan 05 to Dec 09**	**£19.95**
PE3539:	**Jan 06 to Dec 10**	**£19.95**
PE3640:	**Jan 07 to Dec 11**	**£19.95**
PE3741:	**Jan 08 to Dec 12**	**£19.95**
PE3842:	**Jan 09 to Dec 13**	**£24.95**
PE3943:	**Jan 10 to Dec 14**	**£24.95**
PE4044:	**Jan 11 to Dec 15**	**£29.95**
PE4145:	**Jan 12 to Dec 16**	**£29.95**
PE4246:	**Jan 13 to Dec 17**	**£32.95**
PE4347:	**Jan 14 to Dec 18**	**£34.95**

20 YEARS OF PE!

AMAZING BUNDLE OFFER
Every issue from 1999 – 2018

240 magazines for just

£49.95

PE2832: Jan 99 to Dec 03
PLUS!
PE3337: Jan 04 to Dec 08
PLUS!
PE3842: Jan 09 to Dec 13
PLUS!
PE3447: Jan 14 to Dec 18

Do you have the 15-year bundle?

Upgrade to the new 20-year bundle for just £9.95 with: PE15to20

Download your collections at: www.electronpublishing.com

Practical Electronics

Volume 49. No. 8
August 2020
ISSN 2632 573X

Editorial offices
Practical Electronics **Tel** 01273 777619
Electron Publishing Limited **Mob** 07973 518682
1 Buckingham Road **Fax** 01202 843233
Brighton **Email** pe@electronpublishing.com
East Sussex BN1 3RA **Web** www.epemag.com

Advertisement offices
Practical Electronics Adverts **Tel** 01273 777619
1 Buckingham Road **Mob** 07973 518682
Brighton **Email** pe@electronpublishing.com
East Sussex BN1 3RA

Editor	Matt Pulzer		
General Manager	Louisa Pulzer		
Digital subscriptions	Stewart Kearn	**Tel**	01202 880299
Online Editor	Alan Winstanley		
Web Systems	Kris Thain		
Publisher	Matt Pulzer		

Print subscriptions
Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH **Tel** 01202 087631
United Kingdom **Email** pesubs@selectps.com

Technical enquiries

We regret technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine We cannot provide data or answer queries on articles or projects that are more than five years old

Questions about articles or projects should be sent to the editor by email: pe@electronpublishing.com

Projects and circuits

All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

A number of projects and circuits published in *Practical Electronics* employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

Component supplies

We do not supply electronic components or kits for building the projects featured, these can be supplied by advertisers. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

Advertisements

Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

Transmitters/bugs/telephone equipment

We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment as a fine confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

Editorial

At last... back issue downloads!

Practical Electronics has a very long history – so long that we published our golden anniversary issue back in 2014 (do read Alan Winstanley's excellent review at: **www.epe-magazine.co.uk**). We therefore have an impressive back catalogue, and all of the last 20 or so years are available in PDF format on CD/DVD from our online shop. Unfortunately, though, these CDs have never sold particularly well.

The poor sales bothered me, because I knew the content was good, but I always seemed too busy with the current issue to pay much attention to this conundrum. However, a couple of recent orders nudged me in the right direction, and I decided to try and fix the problem once and for all.

First, a reader ordered a couple of CDs to cover a year from the early 2000s, which worked out at around £33. Second, we had a request for more early copies from a South African reader – 'but', he asked, 'is there a download system? The post from Britain seems to take ages at the moment, thanks to Covid-19?'

Our South African reader was not the first to ask for this and now seems to be the perfect time to launch such a service. Downloads offer many advantages. It's instantaneous – you pay and access your issues immediately. There are no postage costs. If you lose your disc then you may well lose your issues, but not with downloads. If your PC dies or your hard disk becomes corrupted, then you can always download again without charge.

Having decided to offer single issues it made perfect sense to offer the same service for our popular 12-month, five-year and 15-year bundles. Except now we have upgraded the 15-year bundle to 20 years. Also, if you've already purchased a 15-year bundle then you can upgrade to the 20-year version for just under a tenner!

Last, but not least, we have slashed the cost of early years – from £33 to just £6.95. We have stopped selling the 6-month CDs and now you can download whole years and at a *much* cheaper price. See the page opposite for details.

Incidentally, the two readers mentioned above have each been given a 20-year bundle for providing the inspiration for these innovations – thank you both.

So, that's your summer reading sorted – just head on over to our online shop and grab some real bargains.

Summer Sale

But wait, there's more! Please see the ad on page 3 announcing our Summer Sale. There are lots of PCB bargains for those of you looking for summer projects.

Keep soldering, and do please keep well

Matt Pulzer

Publisher

The Fox Report

Barry Fox's technology column

New tech norm

We hear a lot about the 'new normal' that will follow the CV lockdown. But even BC (Before Covid), the face of tech journalism was changing radically.

I am old enough to remember when 'official' departments announced new radio, TV and telecoms licensing schemes and services at press conferences where government ministers and their technical advisers explained the tech and answered journalists' questions. The events were often held in government buildings. Heightened risks of terrorism brought tighter security. Eventually, these were replaced by press releases with a phone or fax number to contact for 'further information' – which often came late and lightweight.

The audio, video and telecoms manufacturers continued to hold press conferences, often tied to trade days at public exhibitions, and hosted press visits to their headquarters and labs in far-off lands. Readers must have grown weary of thinly disguised 'thank-you letter' articles about production lines.

The end of the Hi-Fi and video boom years left audio and video companies tightening their belts, and only the consumer electronics giants and computer games and cellphone companies had money to burn on jollies for journalists.

Gadget writers

Meanwhile, the tabloid press – and free newspapers – were breeding a new genus of writers who describe themselves as 'tech specialists' but are better by-lined as 'gadget writers'; they know nothing about how things work and 'review' new devices with a pretty picture, ballpark street price and a few words of off-the-shelf enthusiasm.

I once went to a press event for which the host company had hired a young gadget writer as presenter. One device she showed was a USB-powered cooler. 'Does it rely on the Peltier effect,' we asked. 'No,' she said. 'It's USB.'

'Influencers'

In recent years the gadget pages have been overtaken by online 'tech' sites hosted by a new breed of commentator – the 'influencer'. A colourful character with a pretty face enthuses over new boys' toys, with a superficial hands-on demo after the bizarre ritual now known as 'unboxing'. These sites are two a penny and free to view. The hosts earn money from click-through purchase links and get to keep, and privately flog off, their free product samples.

The manufacturers now often stage party-style launch events with celebrity guests and no opportunity for inquiring journalists to share questions and answers in an open forum Q&A. Influencers no longer need attend unless they fancy a party, because free samples will be sent by courier to any influencer who can prove that a lot of people hit their site. The only loser is the real-world customer who has no way of learning anything of significance about performance or reliability.

An old-school marketing man, who works for one of the world's largest electronics companies, confided recently: 'I was astonished to find that one of our marketing team is now giving a new 65-inch OLED TV, worth £3000, to anyone with 50,000 social media followers. All they have to do is post a picture of them with the set.'

Everything online

I recently signed into a 'New Influencer Ecosystem seminar', organised by a 'new media group' for 'Influencer market agencies', which are the modern online equivalent of the advertising agencies, which traditionally placed glossy adverts in magazines. Most of the seminar speakers spouted meaningless marketing nonsense, but a couple of more honest articulates revealed how the technology now routinely built into computer equipment has been shockingly abused; facial recognition, location detection and language analysis help an influencer gather valuable personal data on 'followers' who 'engage'. The tools are the same as those used to influence elections and referendums. Influencers use click-baiting to temporarily boost their follower numbers by getting people to click on the offer of a free iPad.

The CV crisis has of course put a stop to all physical press conferences and trips. Everything is now being done on line and it is hard to see how

HAMMOND MANUFACTURING®

Ventilated enclosures

hammfg.com/electronics/small-case/plastic/1551v

More than 5000 different enclosure styles:
hammfg.com/electronics/small-case

01256 812812
sales@hammond-electronics.co.uk

and when there can be a return to the old norm of physical events, hands-on demonstrations, viewing and listening sessions, big trade shows and lots of air miles. It's very likely that many manufacturers will see this as a blessed relief from the cost and effort involved in hosting live events.

Sometimes, as with lockdown webinars organised by satellite company SES/Astra and the UK's DTG (Digital Television Group), online events are useful for sharing market research and hard technical facts. For instance, the DTG cited BT's measurement of lockdown daytime Internet traffic in the UK as averaging 5Tbps, with the infrastructure coping well with peaks of 17.5Tbps.

Content-free content

But other online events have presented a worrying view of the likely future. Xiaomi, the Chinese company which claims to be the third-largest smartphone producer in the world, lockdown-launched a new phone with a livestream online 'watch party'. Ahead of the event Xiaomi claimed, 'the same technology as the likes of Apple – for a much more affordable price' with a '£179 phone to rival the £419 iPhone SE.' A bold claim – it takes a lot more than price to get iPhone users to switch to Android.

I signed up and tuned in to the Xiaomi conference on Twitter. The event turned out to be a half-hour pre-packaged promo, with a 15-minute forced wait before a few zippy presentations, some unboxing with next to no hard tech and no opportunity to ask questions and get answers. I asked to try a phone for *PE*, but none is yet available for us.

The link may still be active at **https://bit.ly/pe-aug20-xia**. If so, you can judge for yourself whether this is a good way forward for product launches, or a giant step backwards for serious journalism.

The good old days

I have resignedly regarded CV lockdown as Nature's way of telling me to sort through a lifetime of paper accumulated when researching stories. Among a mass of tedious documents on topics like computer error messages, spectrum allocation, energy policy and changes to UK telephone numbering, I found a note from mid-1992. Philips took a group of journalists from all over Europe to Paris for a one-day in-and-out briefing on the now-forgotten CD Interactive system that was then in the pipeline to launch. As a 'treat', we were taken to the then-new Disney attraction. This is the preface note I wrote to my editor in New York, verbatim:

Check in at Heathrow 5.45am. The coach meeting us at de Gaulle airport fails to turn up. Go by taxi to La Defense. But no-one from Philips is at the other end to pay the taxi drivers. Find way to the roof of the Grand Arche, looking for Philips people while the taxi drivers hold passengers hostage. Find someone from Philips, and go back down again...

Travel to Euro Disney by coach, but cannot leave cases and press kits etc. in coach because it is not waiting.

Walk round Euro Disney carrying bags for an hour or so (time for two rides). Walk miles back to coach to the airport to find that it has left, taking two Italians to the airport and stranding 14 Brits.

Find train station at Euro Disney which has one of three ticket office windows open and a queue of hundreds. No one anywhere to ask advice, so take train to Paris and, mainly by luck, get connection to de Gaulle airport. Run with bags to coach, run round de Gaulle airport and get last flight home only because take-off has been delayed by computer failure.

Barry Fox, FBKS (Fellow, International Moving Image Society)

Fluke/Philips PM3092 Oscilloscope
2+2 Channel 200MHz Delay TB,
Autoset etc – £250

STEWART OF READING
17A King Street, Mortimer, near Reading, RG7 3RS
Telephone: 0118 933 1111 Fax: 0118 933 2375
USED ELECTRONIC TEST EQUIPMENT
Check website **www.stewart-of-reading.co.uk**
(ALL PRICES PLUS CARRIAGE & VAT)
Please check availability before ordering or calling in

HP 54600B Oscilloscope
Analogue/Digital Dual Trace 100MHz
Only £75, with accessories £125

LAMBDA GENESYS	PSU GEN100-15 100V 15A Boxed As New	£400
LAMBDA GENESYS	PSU GEN50-30 50V 30A	£400
IFR 2025	Signal Generator 9kHz – 2.51GHz Opt 04/11	£900
IFR 2948B	Communication Service Monitor Opts 03/25 Avionics	POA
IFR 6843	Microwave Systems Analyser 10MHz – 20GHz	POA
R&S APN62	Syn Function Generator 1Hz – 260kHz	£295
Agilent 8712ET	RF Network Analyser 300kHz – 1300MHz	POA
HP8903A/B	Audio Analyser	£750 – £950
HP8757D	Scaler Network Analyser	POA
HP3325A	Synthesised Function Generator	£195
HP3561A	Dynamic Signal Analyser	£650
HP6032A	PSU 0-60V 0-50A 1000W	£750
HP6622A	PSU 0-20V 4A Twice or 0-50V 2A Twice	£350
HP6624A	PSU 4 Outputs	£400
HP6632B	PSU 0-20V 0-5A	£195
HP6644A	PSU 0-60V 3.5A	£400
HP6654A	PSU 0-60V 0-9A	£500
HP8341A	Synthesised Sweep Generator 10MHz – 20GHz	£2,000
HP83630A	Synthesised Sweeper 10MHz – 26.5 GHz	POA
HP83624A	Synthesised Sweeper 2 – 20GHz	POA
HP8484A	Power Sensor 0.01-18GHz 3nW-10μW	£75
HP8560E	Spectrum Analyser Synthesised 30Hz – 2.9GHz	£1,750
HP8563A	Spectrum Analyser Synthesised 9kHz – 22GHz	£2,250
HP8566B	Spectrum Analsyer 100Hz – 22GHz	£1,200
HP8662A	RF Generator 10kHz – 1280MHz	£750
Marconi 2022E	Synthesised AM/FM Signal Generator 10kHz – 1.01GHz	£325
Marconi 2024	Synthesised Signal Generator 9kHz – 2.4GHz	£800
Marconi 2030	Synthesised Signal Generator 10kHz – 1.35GHz	£750
Marconi 2023A	Signal Generator 9kHz – 1.2GHz	£700

Marconi 2305	Modulation Meter	£250
Marconi 2440	Counter 20GHz	£295
Marconi 2945/A/B	Communications Test Set Various Options	POA
Marconi 2955	Radio Communications Test Set	£595
Marconi 2955A	Radio Communications Test Set	£725
Marconi 2955B	Radio Communications Test Set	£800
Marconi 6200	Microwave Test Set	£1,500
Marconi 6200A	Microwave Test Set 10MHz – 20GHz	£1,950
Marconi 6200B	Microwave Test Set	£2,300
Marconi 6960B	Power Meter with 6910 sensor	£295
Tektronix TDS3052B	Oscilloscope 500MHz 2.5GS/s	£1,250
Tektronix TDS3032	Oscilloscope 300MHz 2.5GS/s	£995
Tektronix TDS3012	Oscilloscope 2 Channel 100MHz 1.25GS/s	£450
Tektronix 2430A	Oscilloscope Dual Trace 150MHz 100MS/s	£350
Tektronix 2465B	Oscilloscope 4 Channel 400MHz	£600
Farnell AP60/50	PSU 0-60V 0-50A 1kW Switch Mode	£300
Farnell XA35/2T	PSU 0-35V 0-2A Twice Digital	£75
Farnell AP100-90	Power Supply 100V 90A	£900
Farnell LF1	Sine/Sq Oscillator 10Hz – 1MHz	£45
Racal 1991	Counter/Timer 160MHz 9 Digit	£150
Racal 2101	Counter 20GHz LED	£295
Racal 9300	True RMS Millivoltmeter 5Hz – 20MHz etc	£45
Racal 9300B	As 9300	£75
Solartron 7150/PLUS	6½ Digit DMM True RMS IEEE	£65/£75
Solatron 1253	Gain Phase Analyser 1mHz – 20kHz	£600
Solartron SI 1255	HF Frequency Response Analyser	POA
Tasakago TM035-2	PSU 0-35V 0-2A 2 Meters	£30
Thurlby PL320QMD	PSU 0-30V 0-2A Twice	£160 – £200
Thurlby TG210	Function Generator 0.002-2MHz TTL etc Kenwood Badged	£65

HP/Agilent HP 34401A Digital
Multimeter 6½ Digit £325 – £375

HP33120A	Function Generator 100 microHz – 15MHz	£350
HP53131A	Universal Counter 3GHz Boxed unused	£600
HP53131A	Universal Counter 225MHz	£350
Audio Precision	SYS2712 Audio Analyser – in original box	POA
Datron 4708	Autocal Multifunction Standard	POA
Druck DPI 515	Pressure Calibrator/Controller	£400
Datron 1081	Autocal Standards Multimeter	POA
ENI 325LA	RF Power Amplifier 250kHz – 150MHz 25W 50dB	POA
Keithley 228	Voltage/Current Source	POA
Time 9818	DC Current & Voltage Calibrator	POA

Marconi 2955B Radio
Communications Test Set – £800

Techno Talk

The benefits of hindsight

Mark Nelson

We ignore the importance of hindsight – and foresight – at our peril. Hindsight doesn't deliver all the answers, but it can shine a useful spotlight on vital information and insights that might not otherwise be obvious. Applying hindsight, this article may help you assess some past predictions in a new light.

Of course, the greatest value of hindsight is that it provides the purest form of 20:20 vision. This is in total contrast to foresight, which is notoriously hazy, no matter what policy leaders may claim. This key shortcoming was best summed up by Danish scientist and Nobel Prize winner Niels Bohr, who famously stated: 'Prediction is very difficult, especially if it's about the future.' Examining hindsight, on the other hand, often provides clues to the best course to follow in the future. It also gives us the ability to recognise truly far-sighted vision, albeit long after the event. Two examples demonstrate how.

The Internet vision of 1942

Here's a report from the May 1942 issue of trade magazine *Electrical Trading*. In the middle of the Second World War, radio pioneer and first chief engineer of the BBC, Peter Eckersley offered his opinions on 'The Future of Radio Communication' when he addressed members of the British Institution of Radio Engineers at their April meeting. He described the limitation of radio communication channels currently available and in explaining how 'wired wireless' alias 'radio relay' could be developed to advantage, he accurately predicted Internet Radio.

'Dealing with wired broadcasting, Captain Eckersley thought this system would provide a solution to ether congestion, and envisaged a future when perhaps a special cable would be laid to every house, not only in this country, but in every country of the world, linking continents as far apart as Europe and America, although he realised the present difficulties of operating a submarine cable of such dimensions. The number of channels available in a wired system would be infinite, and in this he saw in the future a solution of our broadcasting problems.'

When Eckersley spoke, many towns in Britain already had radio relay systems delivering a choice of interference-free radio programmes by cable. David Elliott recalls: 'My parents were on radio relay in Chelsea in London in October 1939. They paid seven shillings for the loudspeaker, which they then owned. It was wired up to the relay system, for which they paid around one shilling a week. There were two programmes, the BBC Home Service and later BBC Forces. In 1945, The Forces service became BBC Light Programme. In 1949 two more stations were added: BBC Third Programme and a fourth station which provided popular music from abroad and Radio Luxembourg in the evening. In 1955, BBC and ITV television were added.' More memories of wired wireless at: **https://bit.ly/pe-aug20-relay**

Cellphones foreseen in 1946

Today, the name of prolific author Miles Henslow is little remembered, although you may have come across the annual *Hi-Fi Yearbook* publications that he produced during the 1950s and 60s. In 1946, he wrote a comprehensive book titled *The Miracle of Radio* that is a detailed survey of how wireless contributed to the Allied success in World War Two. What caught my eye is the uncanny accuracy of his prediction of how cellular radio might develop, at a time when the only mobile radio equipment in day-to-day use was in selected police cars (the radio apparatus completely filled the boot of a squad car).

'Maybe it will sound a far-fetched idea today, but the time is surely approaching when everyone will be able to carry about with him a small radio telephone. War-time development of apparatus to work on very short wavelengths has opened up many new entrancing possibilities. Hundreds of thousands of 'radio-telephone channels' can be used over short distances without interference; and the installation of a network of automatic telephone exchanges might well be utilised for handling the calls from a multitude of pedestrian or automobile telephone subscribers, to sort them out and to pass them by line – or radio link – to main exchanges. Certainly, it is but a matter of time before the railway traveller is able to pick up the 'phone and dial his office or his home.'

The matter of time was in fact 39 years; Vodafone and BT Cellnet introduced cellular mobile radio in 1985. The hand-portable phones used were the Motorola 'brick', costing around £1,000 in 1985 money (equivalent to £2,630 or £3,050 today, according to which inflation calculator you use).

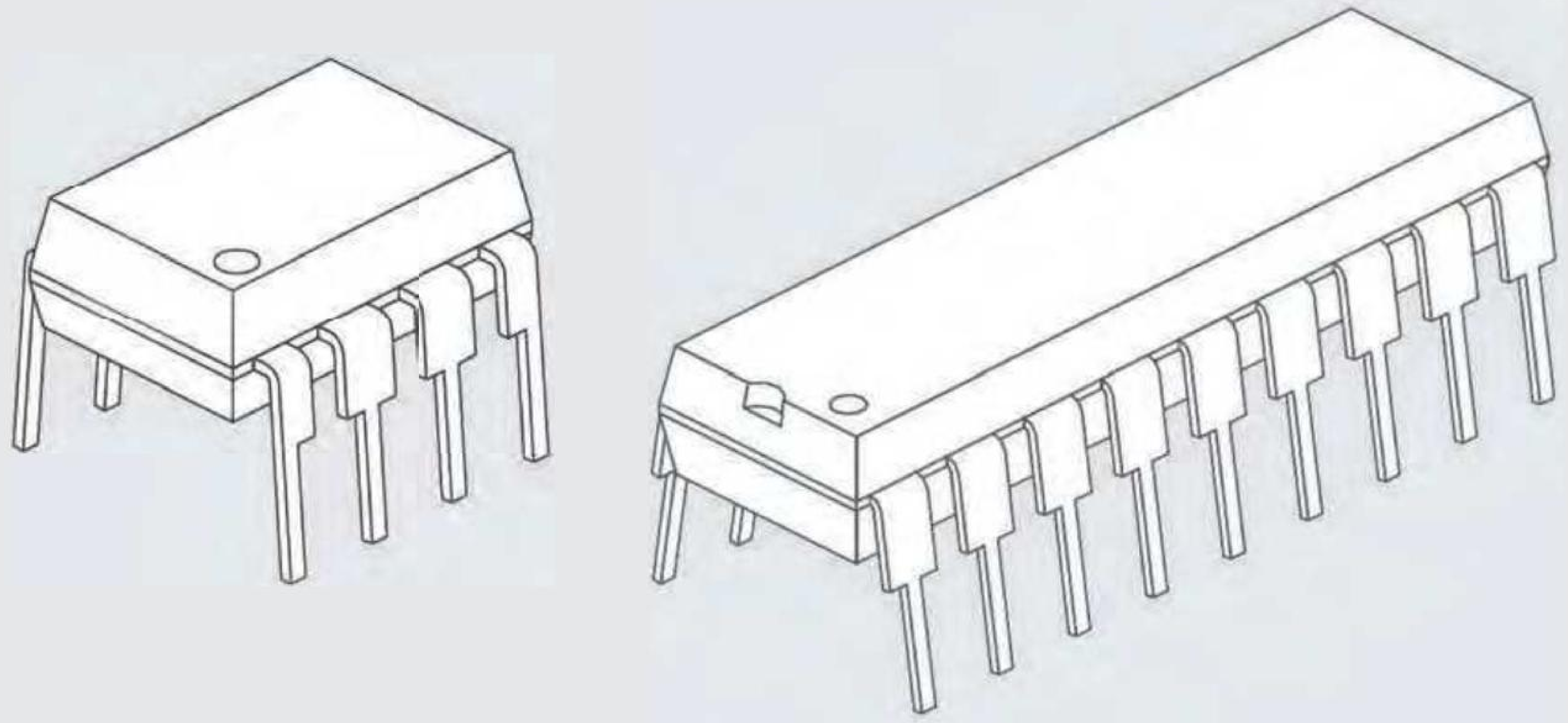
Correcting errors

Although hindsight doesn't give us the ability to alter past mistakes, it does enable us to acknowledge and correct the blunders that we've made – and I made an egregious error in the June issue. I stated that, from memory, the first S-DeC breadboards cost about £20, equivalent to £266 in today's money.

What utter tosh! The actual cost was 29s 6d (plus 6d for post and packing). This equates to £22.93 according to the inflation calculator that I used. How could I get it so utterly wrong? I blame the fuzzy logic in my befuddled brain – but I do have an excuse. What I was remembering, poorly expressed, was its equivalent price and the £20 I stated is not far off the £22.93 in 2020 money that the breadboard would have cost me in 1967. In those days, I was a bus conductor, earning £6 a week. Thirty shillings was a quarter of my weekly pay packet, so you can see why I couldn't afford to fork out that much money for an S-DeC!

If you're wondering how I suddenly developed this miraculous feat of total recall, I'll tell you. I was shifting a pile of paperwork last week and tucked under this I discovered a long-forgotten October 1967 copy of *Practical Electronics*. Back then I never dreamed I would one day write for the publication. In this issue was a full-page advertisement introducing S-Dec as 'the breadboard for the transistor age'. We have come a long way since then, haven't we?

Need a programmed PIC for your Practical Electronics project?



Now available from the Practical Electronics online shop at:
www.electronpublishing.com

Teach-In 8 CD-ROM
Exploring the Arduino

This CD-ROM version of the exciting and popular *Teach-In 8* series has been designed for electronics enthusiasts who want to get to grips with the inexpensive, immensely popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. *Teach-In 8* provides a one-stop source of ideas and practical information.

The Arduino offers a remarkably effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle wirelessly or via the Internet. *Teach-In 8* is based around a series of practical projects with plenty of information for customisation. The projects can be combined together in many different ways in order to build more complex systems that can be used to solve a wide variety of home automation and environmental monitoring problems. The series includes topics such as RF technology, wireless networking and remote web access.

FREE CD-ROM
£8.99
ELECTRONICS TEACH-IN 8
FREE CD-ROM SOFTWARE FOR THE TEACH-IN 8 SERIES
INTRODUCING THE ARDUINO
• Hardware – learn about components and circuits
• Programming – powerful integrated development system
• Microcontrollers – understand control operations
• Communications – connect to PCs and other Arduinos
Teach-In 8
PLUS...
PIC n'MIX

PRICE
£8.99
Includes P&P to UK if ordered direct from us

PLUS: PICs and the PICkit 3 – A beginners guide

The CD-ROM also includes a bonus – an extra 12-part series based around the popular PIC microcontroller, explaining how to build PIC-based systems.

SOFTWARE

The CD-ROM contains the software for both the *Teach-In 8* and PICkit 3 series.

ORDER YOUR COPY TODAY!
JUST CALL 01202 880299 OR VISIT www.epemag.com

Net Work

Alan Winstanley

In barely a dozen weeks we've entered a different world. This month, *Net* Work looks at the accelerated rise of video conference calls and Yubico's latest Security Key.

At the time of writing, there are some welcome signs that the 'lockdown' is starting to ease, as authorities grapple with the problem of allowing our socially distanced lives to return to normality once again. In these strange and surreal times, our society has, thanks largely to Internet access, adapted and continued functioning even when we're forced to stay indoors and avoid physical contact with anyone else. Whether we like it or not, the current crisis is smashing us into doing things differently – as a timely example, just when I was typing this month's copy I also had a doctor's consultation via my mobile phone camera: video meetings that were once unthinkable are destined to become normal.

Amazon's Echo Show desktop LCD has a built-in camera for video calling, messaging your contacts or checking your Blink security cameras. Chatting via Facebook-owned Whatsapp, Microsoft Skype or Apple's FaceTime is now second nature for many. Facebook continues to push the latest version of its Portal hardware (see: *Net Work*, February 2020) at families, as a user-friendly way of making widescreen video calls. Back in May, Facebook also launched Messenger 'rooms' which accommodate groups of up to 50 users, and Google Meet fired back with a revamp that made it feel less 'corporate'. Google Meet is free for individuals (up to 100 per session) and business plans are also available. Launched in 2017, Microsoft has been heavily promoting Teams, their own 'chat-based workspace' for collaborative networking, while video-conferencing provider Zoom now faces the classic dilemma of satisfying those who pay nothing for the service anyway, since profits lie with their business users instead. Education has also seen some action: in Britain, many educators tackled the challenge of teaching pupils online for the first time using a choice of these tools, akin to launching Australia's *School of the Air* (*Net Work*, April 2019). With our unease about video calling now overcome, the idea of talking by video has come of age.

Amazon's Echo Show is a desktop LCD available with 5.5, 8 and 10-inch screens.

Camera, action

While tablets and laptops often have built-in cameras for making video calls, it's been a bit tougher for some desktop PC owners wanting to embrace video calling during the lockdown. Retail stocks of PC webcams flew out the door, leaving only anonymous unbranded webcams stuck on the shelf. Needing one to make a Skype call to some locked-down relatives, the author eventually resurrected an HD Logitech Webcam Pro 9000 for his PC, the same one that readers first saw back in February 2012, when it cost £35 – used ones now fetch double or triple that on auction websites. After the usual installation and software hassles, and against all the odds, it finally installed successfully and works well in Skype, thanks mainly to finding an earlier (and more compatible) driver buried on a hard disk. You can download Skype for Windows (from: **www.skype.com**). On mobile devices (iPad, Android and Kindle Fire HD), check the usual app stores.

The digital motion tracking on this legacy webcam also works well: if I move, the image follows me, though colour rendering is very odd under night-time LED lighting. Video-calling has become all the rage with TV news programmes and government meetings. Fortunately for participants, popular video chat software can recognise a human face and electronically 'blur' the background or swap it altogether for another JPEG.

We sometimes lose track of how far this digital technology has advanced in the past 10 or 15 years, starting with static ANPR cameras picking out car licence plates, fingerprint scanning, facial recognition and now even AI-driven lip-reading (*Net Work*, November 2019). With developments in AI racing ahead, autonomous pre-programmed machines and their sensory systems are being entrusted with the interpolation of data, 'deep-learning' and taking decisions themselves or even driving cars automatically; often, all but eliminating human intervention from the proceedings. As machines become more 'intelligent', this trend towards utilising human-crafted AI will dominate many areas of our lives in years to come.

Speed limits

When checking broadband deals to help out a new neighbour, I found the market for residential services seemed to be hardening, with consumers facing less choice from the handful of ISPs that now control most of the sector. When trying to source 'basic' ADSL/ADSL2 broadband in my postcode, several telcos stated it was 'not available' – but were happy to quote for fibre broadband tariffs instead. I also noticed some confusion over naming conventions. 'Superfast' broadband (say 30-50Mbps or more) typically delivers fibre Internet to a roadside cabinet, with the telephone cable wire carrying the last leg. So-called 'Faster' broadband may refer to legacy ADSL2+ (say up to 20 Mbps or more – depending on distance over the phone lines), which may still be used today by customers having modest demands. (In some locations it's all that might be available.) But I found that even when searching for the cheapest tariffs, superfast fibre was the only choice on offer: BT suggested 'Fibre Essentials' with a 25Mb guarantee, or 50Mb 'Fibre 1'. Obviously, the network is moving to

Yubico's USB Security Key adds physical 'fingertipping' verification to web services: check hardware and service compatibility before investing.

migrate users onto fibre broadband as a matter of course, and prices are levelling up across the board.

Coming next is 'Fibre to the Premises' (FTTP) with fibre Internet streaming directly to your property. Some small independent UK ISPs are starting to roll this out, including Internetty (**www.internetty.uk**). They call their 250Mb fibre 'Ultrafast' (£32 per month) and 1Gb is dubbed 'Hyper-fast' (£45). Cable operator Virgin Media delivers its own 'superfast' broadband over its own cables and, commendably, they avoid the use of meaningless product names. You can check whether they operate in your area by searching your postcode on: **www.virginmedia.com**

When shopping for broadband, buyers might need to factor in line rental and the cost of landline calls, or the ISP may bundle in TV or mobile calls too. 'Broadband only' deals with no phone are available, but do check contract lengths, availability of free routers (delivery extra) and exit fees if you move house. At the time of writing, TalkTalk charges £60 to leave. Most of all, double check the speeds and don't be swayed by confusing marketing jargon!

Yubico Security Key: fingertap control

Previously, I covered the use of Two-Factor Authentication (2FA) as an extra security step to safeguard your social media, shopping and email accounts against unauthorised access. Instead of receiving SMS texts containing PIN numbers, so-called 'hardware tokens' such as the Yubico range of USB keys (see previous months) simply need a fingertap to verify your identity and they work with an expanding selection of online services (see: **www.yubico.com/setup/compatible-services**).

Intellivision Amico launches this October and promises reimagined video gaming fun for family and friends.
(Image: Intellivision, YouTube)

Yubico kindly sent *Practical Electronics* a USB Security Key, a slim but tough plastic USB device that fits an ordinary Type-A USB port found on desktops and laptops. It was easily set up with Facebook: after typing in logins as normal on a PC the security key's gold touchpad flashed, inviting a fingertap to complete the process. It worked flawlessly and I logged in on both W7 and W10 PCs without a hitch. Mobile devices probably have a USB-C port instead, potentially a job for the diminutive YubiKey 5C (also supplied). Unfortunately, I found Facebook's mobile app did not recognise hardware tokens in the first place, which shows how disjointed this evolving market currently is. Apart from that, if USB is a problem then NFC USB Security Keys are sold, and an NFC version of the 5C is in the pipeline. Users should consider what USB form factors meet their needs, or whether NFC is usable instead, to get the best out of these security keys. NFC versions are likely to prove more futureproof.

Yubico lists setup instructions for a range of web services, including Google, Microsoft Accounts and Azure Active Directory, and UK Government Verify (but *not* Gateway). For everyday Internet users, it's still relatively early days for USB hardware tokens as website operators edge towards integrating this added security into their own systems. At the moment, and taking Facebook as an example, a Yubico key makes access more secure, but not faster. However, single-factor (passwordless) standalone operation, which requires no username or password, is promised by Yubico as the FIDO2 protocol rolls out. Gradually, it should become possible to log in directly into services with just a single touch tap. Yubico security products are available direct from Yubico.com and Amazon.

Intellivision lives!

The SpaceX Starship is aiming for the moon, then Mars, with ambitious plans to build a moon base.

Now something for retro videogaming fans. Old hands from the 1980s will remember the Mattel Intellivision games console, which in 1979 was the first 16-bit video game system to hit the market. Its 'D' pad and CRT-friendly screensaver were ahead of their time. Mattel Electronics introduced the first speech synthesis unit (the Intellivoice), a music synthesiser keyboard and was first with game downloads, via a cable subscription. Mattel had grand plans for their console, but some ambitious add-ons were poorly supported and quietly dropped. A former engineer posted that he worked on an adaptor (never launched) that would have enabled Intellivisions to handle simple online banking – 40 years ago! Mattel Electronics then dabbled disastrously in the nascent home computer market, following their ill-fated Aquarius computer in 1983: a modem and futuristic home automation control using X10 were pencilled in, but the Aquarius barely made it off the drawing board. After a lot of bloodletting in the 1980s retail market, Intellivision disappeared when both Nintendo and proper home computers arrived on the scene.

The excellent news for Intellivision fans is the promised launch of a smart new console – the Intellivision Amico – with HDMI and touchscreen controllers that are fit for the 21st century. The Amico hopes to disrupt the market and it promises 'clean fun for family and friends' like it used to be, rather than being for 'hard core gamers' playing in isolation. Shunning the complexity of 3D games, many familiar old titles will be re-imagined and ported onto the new system (more details at: **www.intellivisionamico.com**). The launch date is 10 October 2020, in good time for Christmas, and I can't wait!

In comparison, a forthcoming new Atari-badged console has suffered from nothing but endless delays and an excoriating commentary seen on *The Register* makes for very painful reading. More at: **https://tinyurl.com/y9u3nsu5**

The Space Race

June was an exciting month for space program fans, starting with a commercial SpaceX capsule docking with the international space station. Frustratingly, though, the vertical landing of its re-usable booster stage onto a drone ship was a feat lost on TV news presenters. Fortunately, an accurate (and fun) ISS crew docking simulator was put online for surfers to try, see: **https://iss-sim.spacex.com**

Another 60 SpaceX Starlink satellites were deployed in June, bringing the total to nearly 500 with possibly tens of thousands more slated for the future. One satellite has a test pop-up sun visor designed to block dazzling reflections from the spacecraft, in an effort to appease astronomers. Still undergoing tests is Starship, their heavy lifter aiming for the moon, Mars and beyond. Small rival OneWeb (see *Net Work*, May 2019) had launched 74 satellites in its own program, but filed for Chapter 11 bankruptcy protection at the end of March, citing market turbulence and Covid-19 disrupting its finances. It hopes to refocus operations and find a way forward.

In Britain, Scotland-based private venture Skyrora conducted the first successful rocket engine test on UK soil in 50 years and hopes to offer satellite launcher services in 2022. The company is also working hard to inspire and attract tomorrow's engineers. More news on: **www.skyrora.com**

News in brief

Microsoft is fighting back in the browser wars with its Edge web browser, now available in automatic updates for Windows 10. Based on Chromium and rebuilt from the ground up, it promises better compatibility with websites that will 'work as they're supposed to work'. Windows 7+ users can fetch it from: **www.microsoft.com/en-us/edge** and macOS, iOS and Android versions are there too. Edge has been well received, seems very fast in use and is also compatible with third-party Chromium extensions.

The latest in 5G communications has reached Mount Everest, after five 5G base station upgrades were installed by Huawei Technologies near the top of the world's highest mountain. Huawei is also investing in its first new factory outside of China. Dedicated to 4G and 5G production for Europe, the plant is expected to be built in France at a cost of €200m. In the US, America's war on Huawei continues: a bill was passed to rip out and replace 'communications equipment or services posing a national security risk', naming no names. The Secure and Trusted Communications Networks Act of 2019 was signed off earlier this year and may cost America upwards of $800m to implement. Britain's stance on Huawei is said to be changing too, as the UK is now considering ways of sourcing 5G hardware from other manufacturers in Japan and South Korea, in light of political challenges facing the UK. Britain's BT says it would take seven years to eliminate Huawei entirely from its network.

Ransomware has now been found running within virtual machines, reports British security firm Sophos. 'Ragnar Locker ransomware was deployed inside an Oracle VirtualBox Windows XP virtual machine. The attack payload was a 122MB installer with a 282MB virtual image inside – all to conceal a 49kB ransomware executable,' they blogged at: **https://tinyurl.com/y7tk8y3s**

See you next month for more *Net Work!*

The author can be reached at:
alan@epemag.net

GET THE LATEST COPY OF OUR TEACH-IN SERIES AVAILABLE NOW!

Order direct from Electron Publishing

PRICE £8.99

(includes P&P to UK if ordered direct from us)

ELECTRONICS TEACH-IN 9
£8.99
GET TESTING!
FREE CD-ROM
PIC n' Mix
Teach-In 9 Get Testing

Teach-In 9 – Get Testing!

This series of articles provides a broad-based introduction to choosing and using a wide range of test gear, how to get the best out of each item and the pitfalls to avoid. It provides hints and tips on using, and – just as importantly – interpreting the results that you get. The series deals with familiar test gear as well as equipment designed for more specialised applications.

The articles have been designed to have the broadest possible appeal and are applicable to all branches of electronics. The series crosses the boundaries of analogue and digital electronics with applications that span the full range of electronics – from a single-stage transistor amplifier to the most sophisticated microcontroller system. There really is something for everyone!

Each part includes a simple but useful practical test gear project that will build into a handy gadget that will either extend the features, ranges and usability of an existing item of test equipment or that will serve as a stand-alone instrument. We've kept the cost of these projects as low as possible, and most of them can be built for less than £10 (including components, enclosure and circuit board).

FREE COVER-MOUNTED CD-ROM

On the free cover-mounted CD-ROM you will find the software for the *PIC n' Mix* series of articles. Plus the full *Teach-In 2* book – *Using PIC Microcontrollers – A practical introduction* – in PDF format. Also included are Microchip's *MPLAB ICD 4 In-Circuit Debugger User's Guide*; *MPLAB PICkit 4 In-Circuit Debugger Quick Start Guide*; and *MPLAB PICkit4 Debugger User's Guide.*

ORDER YOUR COPY TODAY JUST CALL 01202 880299 OR VISIT www.electronpublishing.com

Win a Microchip J-32 Debug Probe

Practical Electronics is offering its readers the chance to win a Microchip J-32 Debug Probe (DV164232) – and even if you don't win, receive a 20%-off voucher, plus free shipping for one of these products.

The J-32 Debug Probe Debugger/Programmer provides affordable, fast and easy debugging and programming for Microchip's PIC32, and SAM MCU and MPU products.

The J-32 Debug Probe is fully integrated into Microchip's powerful MPLAB X Integrated Development Environment (IDE) and easy-to-use Integrated Programming Environment. Connecting existing hardware is simple as it keeps the SEGGER J-Link/J-Trace 20-pin debugging connector. The J-32 Debug Probe has all the necessary performance and features 32-bit MCU and MPU developers need.

From SEGGER, the J-32 also benefits from the following advanced software utilities, many of which are free:

Worth $200.00 (approx £162.50) each

SystemView (V2.x) provides runtime recording and captures tasks, interrupts, timers, API calls and user events, and provides live analysis of captured information.

J-Scope application displays runtime data of multiple variables in an oscilloscope format. Simply connect the J-32 Debug Probe to your target, program and start J-Scope. Each variable can be individually manipulated with familiar oscilloscope controls and features.

Remote Server allows the J-32 Debug Probe to debug from anywhere in the world. This is especially useful when your software expert needs to resolve development or product issues in another region.

J-Mem permits direct RAM and Special Function Register (SFR) modifications without a bulky Integrated Development Environment. Memory can be displayed in an 8-, 16- or 32-bit format and can be exported to a .bin file. J-Mem is compatible with Windows, macOS and Linux operating systems.

Free-to-enter competition

J-32 Debug Probe
Microchip
Developed by SEGGER
Development Tools
Microchip
J-32 Debug Probe
Microchip
Developed by SEGGER
www.segger.com

Microchip
J-32 Debug Probe

How to enter

For your chance to win a Microchip J-32 Debug Probe or receive a 20% off voucher, including free shipping, enter your details in the online entry form at:

https://page.microchip.com/PE-J-32.html

Closing date

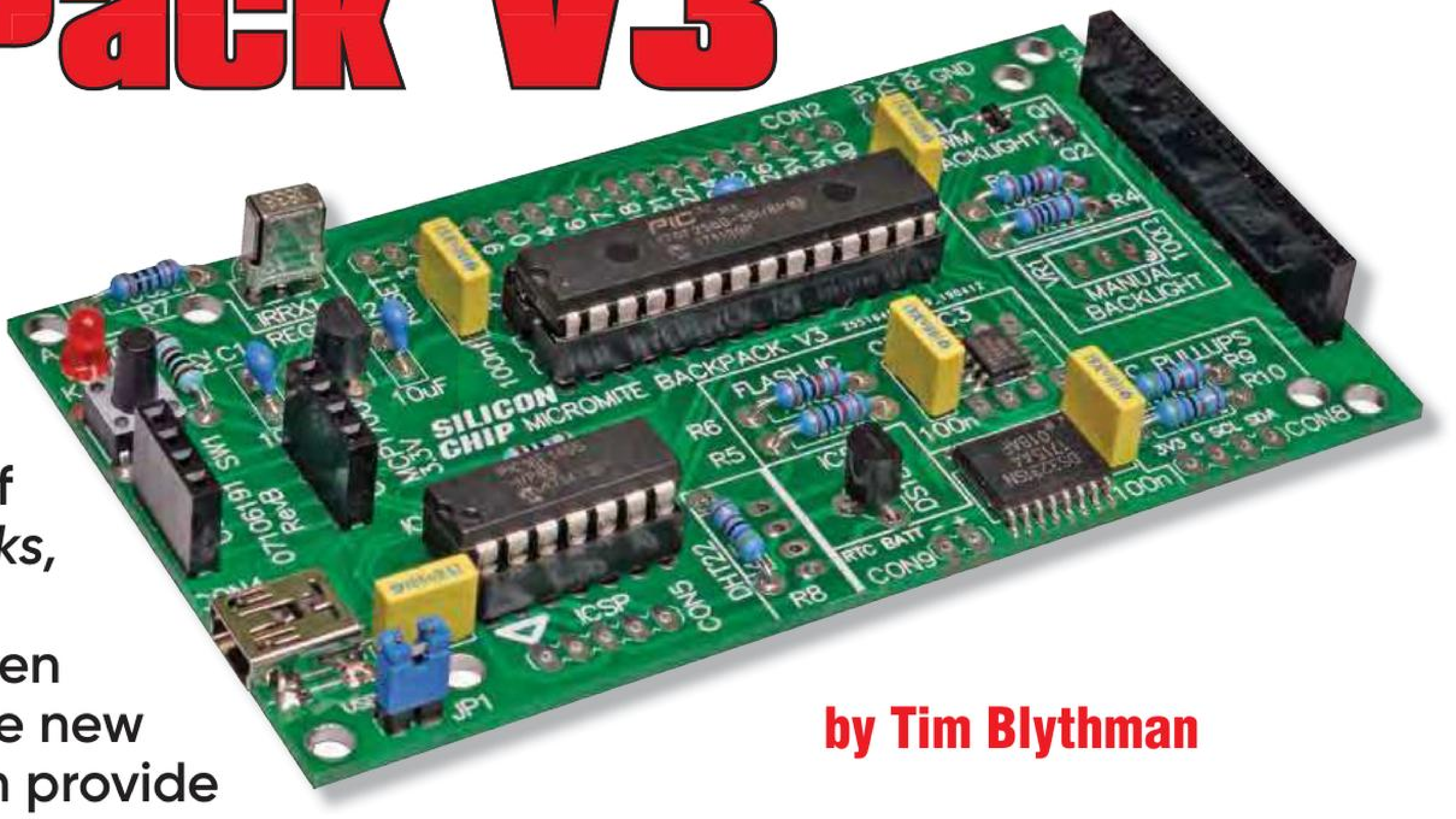
The closing date for this offer is 31 July 2020.

April 2020 winner

David Carney

He won a Microchip Curiosity PIC32MZ EF Dev Board 2.0

Micromite LCD BackPack V3



This *BackPack* is the most convenient and powerful yet.

It has all the features of the *V1* and *V2 BackPacks*, supports both 2.8-inch and 3.5-inch touchscreen displays and boasts five new optional features which provide convenient functions. These include extra memory, temperature, humidity and pressure sensors, a real-time clock, an infrared receiver and more!

by Tim Blythman

In our recent article on 3.5-inch touchscreen displays (see the June 2020 issue), we looked at three different screens. But we were particularly impressed by one. It uses an ILI9488 controller with SPI interface and has the same connections as the popular 2.8-inch touchscreen display used by the original and *V2 BackPacks*.

For that article, we supplied code to drive that new display from an Arduino and a standard Micromite. We also mentioned that we planned to write some CFUNCTIONs to speed it up, as the BASIC code is quite slow at refreshing the screen.

Not only have we now done that, but we've also designed a new version of the *BackPack* to properly accommodate the larger, higher-resolution screen with twice as many pixels as the original (480×320 compared to 320×240).

While this article gives sufficient information for you to fully understand what we've done, if you haven't seen the *V2 BackPack* article in the May 2018 issue of *PE*, you might want to read that before coming back to this article.

Essentially, the *BackPack* is a small PCB that hosts a PIC32 running the Micromite firmware. It also provides a simple power supply, a USB interface, a header and mounting screws for a colour touchscreen and an I/O pin header.

The best part about it is that MMBasic has native touchscreen support. It's such a great idea that we've used the *BackPack* in numerous other projects.

But the *V3 BackPack* is more than just a screen upgrade. While you can build the new *V3 BackPack* using the same components as the *V2 BackPack*, you can also add several extra components to add handy features without needing to connect extra modules, PCBs or wiring.

You can fit it with an infrared receiver/decoder for remote control, a Flash memory IC or SRAM, a DHT22 temperature and humidity sensor, a DS18B20 temperature sensor or a DS3231 real-time clock IC.

There's also a header for connecting additional I^{2}C devices, such as a BMP180/BMP280/BME280 temperature/pressure/humidity sensor, which can be mounted directly to the board if desired.

Also, this *BackPack* lets you use the SD card socket that's mounted on the back of the touchscreen board.

All the functions that were in the original and *V2 BackPack* are retained in the *V3 BackPack*, including its 50MHz 32-bit processor loaded with a powerful BASIC interpreter, which can be programmed over a virtual USB serial port.

Circuit description

We'll start by describing the core functions, which are carried over from the *V2 BackPack*.

Refer to Fig.1, the circuit diagram. IC1 is the main processor which runs the MMBasic interpreter and handles other functions. It is a PIC32MX170F256B (or the 50MHz variant,) in a 28-pin dual inline package. It requires some bypass capacitors for normal operation: two 100nF MKT capacitors across its supply rails and a 10µF ceramic capacitor to filter its internal core supply.

There's also a 10kΩ resistor used as a pull-up on IC1's RESET line, to prevent spurious resets.

IC2 is a Microchip PIC16F1455 microcontroller which is both a USB/serial converter and a PIC32 programmer – the *Microbridge* article in the May 2018 issue of *PE* describes its functions in more detail.

When running as a USB/serial converter, pin 5 on the PIC16F1455 receives data (ie, data from the Micromite to the PC USB interface) and pin 6 transmits data (from the PC USB interface to the Micromite).

These signals also run to the edge pins for the console connection (CON1) in case you build this PCB but for some reason do not plug the *Microbridge* IC, IC2, into its socket. In this case, you can use an external USB/serial converter.

The PIC32 programming interface from the *Microbridge* is on pins 7, 2 and 3 of IC2. These provide the reset function, program data and clock signals, which connect to pins 1, 4 and 5 respectively on the Micromite (IC1).

The programming output on the *Microbridge* is only active when it is in programming mode, so the *Microbridge*

Features

- **Compatible with Micromite LCD BackPack V1 and V2**
- **Suits 2.8-inch and 3.5-inch touchscreen LCD modules**
- **Built-in Microbridge provides serial communications and programming interface**
- **Mini USB socket for power and communication**
- **Native support for 3.5-inch display using initialisation CFUNCTION**
- **Manual or software (PWM) dimming for LCD backlight**
- **4-pin I^{2}C communication header**
- **Optional onboard infrared receiver**
- **Optional onboard DHT22 temperature and humidity sensor or DS18B20 temp sensor**
- **Optional onboard DS3231 real-time clock**
- **Optional onboard Flash memory/RAM IC**
- **Optional onboard BME180/BME280/BMP280 temperature/pressure/altitude sensor**

does not interfere with the Micromite when it is using pins 4 and 5 as general-purpose I/O pins.

Switch S1 is used to select programming mode and LED1 indicates the mode (lit solid when in programming mode).

CON2 is the main I/O connector for the Micromite and is designed so that it can plug into a solderless breadboard for prototyping. The connector also makes it easy to add a third PCB to the LCD *BackPack* 'stack' which can carry circuitry specific to your application (eg, amplifiers or relay drivers). This connector is wired identically to the original *BackPack*.

The Micromite communicates with the LCD panel using an SPI interface where pin 3 on the Micromite feeds data to the LCD while pin 25 provides the clock signal. When the Micromite pulls pin 6 low, it is communicating with the LCD panel, and when pin 7 is pulled low, the Micromite is communicating with the touch controller on the display panel.

The 28-pin Micromite has only one SPI port and so pins 3, 14 and 25 (SPI data and clock) are also made available on CON2 so that you can also use this SPI serial channel to communicate with external devices.

Backlight control

You have two choices for controlling the brightness of the LCD's backlight. The first is to fit MOSFETs Q1 and Q2 to the PCB, along with their associated resistors (this area is marked with a box on the PCB). When you do this, PWM output 2A on the Micromite is used to control the backlight brightness from within your program. This is described in more detail later.

Alternatively, as with the original *BackPack*, you can fit VR1, a 100Ω trimpot. This is in series with the power to the backlight LEDs, so it limits the current drawn by them and therefore sets the brightness. Note that you should install one set of components or the other (not both). You also have the option of fitting a link across VR1's pads to permanently set the backlight to full brightness.

The LCD panel has a 3.9Ω resistor in series with the backlight so you will not burn out the backlight if you set the PWM output to 100%, wind VR1 to zero ohms or link it out.

The power supply is derived from either the 5V connector pin on CON1, or if JP1 is installed, from USB connector CON4. Powering the Micromite LCD *BackPack* from USB power is handy during program development, but for an embedded controller application you would typically remove the jumper from JP1 and supply 5V power via CON1.

Note: do not power the *BackPack* from both CON1 and USB – you could cause damage to the USB interface on your computer.

The 3.3V power supply for both the Micromite and the *Microbridge* is provided by REG1, which is a fixed-output regulator with a low dropout voltage suitable for use with USB power supplies. This supply is also made available on CON2 so you can use it for powering external circuits (to a maximum of 150mA).

Improvements

As mentioned above, one of the major improvements with the *BackPack V3* is that you can use either a 2.8-inch 320×240 pixel touchscreen or a 3.5-inch 480×320 pixel touchscreen. The board is sized to fit the larger screen. It still fits comfortably inside a UB3 jiffy box, the same box which we've used to house several Micromite *BackPack*-based projects over the years.

We have also designed the board so that with both screen options, the onboard SD card socket is wired up to IC1. While the Micromite Plus software has read/write support for SD cards, it will not work on any through-hole PICs. The regular Micromite code, which works on our 28-pin DIP chip, does not natively support SD cards.

However, it would be possible to write BASIC code (or perhaps a CFUNCTION) to access an SD card with the regular Micromite, so we decided to wire up the SD card socket that already exists on the touchscreen module.

This extra header also helps to hold the touchscreen squarely onto the *BackPack* module without needing mounting hardware. The SD card is connected to the same SPI interface that's used to drive the touchscreen, but it has a separate CS line, which is connected to pin 4 on the Micromite. If you don't insert an SD card, it won't have any effect on this pin so it can be used for other purposes.

We decided if we were making these changes then we should add some other useful features at the same time.

Added features

The *BackPack V3* has provision for many extra onboard components which provide various useful functions. None of these need to be fitted; if you leave them off, the board will work much the same as the *V2 BackPack*, except for the option of the larger screen and SD card access. These five optional extra components can be used to add extra features to your Micromite project without needing to add another board or module.

1. 3.3V Infrared receiver (IRD1)

This mounts near the edge of the board, so that its leads can be bent to face outwards for convenient remote control of the unit. Its supply is filtered for reliable operation. Its output is connected to Micromite pin 16, which is the MMBasic IR input pin, making it trivial to receive remote control commands in BASIC.

The IR receiver should ideally be a 3.3V type such as the Vishay TSOP2136 or TSOP2138. However, we tried 4.5V receivers (eg, Jaycar ZD1952) on a 3.3V supply and they normally worked fine.

2. Serial Flash memory or static RAM

Use either an 8-pin DIP or SOIC package for IC3. If you aren't using the SD card interface, you can fit a Flash or SRAM chip with a standard pin-out to the board and use this to store configuration data, logging data or temporary working data. These chips are easier to drive than SD cards; the BASIC code to access them is easy enough, and we provide a sample sketch to do this.

The memory chip's SPI interface is connected to the usual SPI pins on the Micromite, while the chip select line ($\overline{\text{CS}}$, pin 1) goes to pin 4 of IC1, same as for the SD card. That is why you can't use both at the same time.

If fitting this IC, you have the option to fit either or both of the pull-up resistors on pin 3 (write protect/$\overline{\text{WP}}$) and pin 7 ($\overline{\text{HOLD}}$). These may be required to read and write data on the chip. We've also provided for a 100nF supply bypass capacitor; always a good idea.

Ensure your IC's pin-out for this board matches that shown and that it can run off a 3.3V supply. This is by far the most common pin-out for 8-pin serial memory chips and they will virtually all operate from 3.3V, but it's best to check.

3. 4-pin I^{2}C header

This connects to the I^{2}C bus and 3.3V power supply (CON8). Two 4.7kΩ pull-up resistors are provided on the SCL and SDA lines, although these can be left out if pull-ups are provided externally.

The pinout of CON8 matches the commonly available BMP180/BMP280 temperature and atmospheric pressure sensor modules, as well as the BME280 temperature/pressure/humidity module. So any of these can be soldered directly to the *BackPack* at CON8.

Alternatively, a four-way header can be fitted and leads run to one of the many

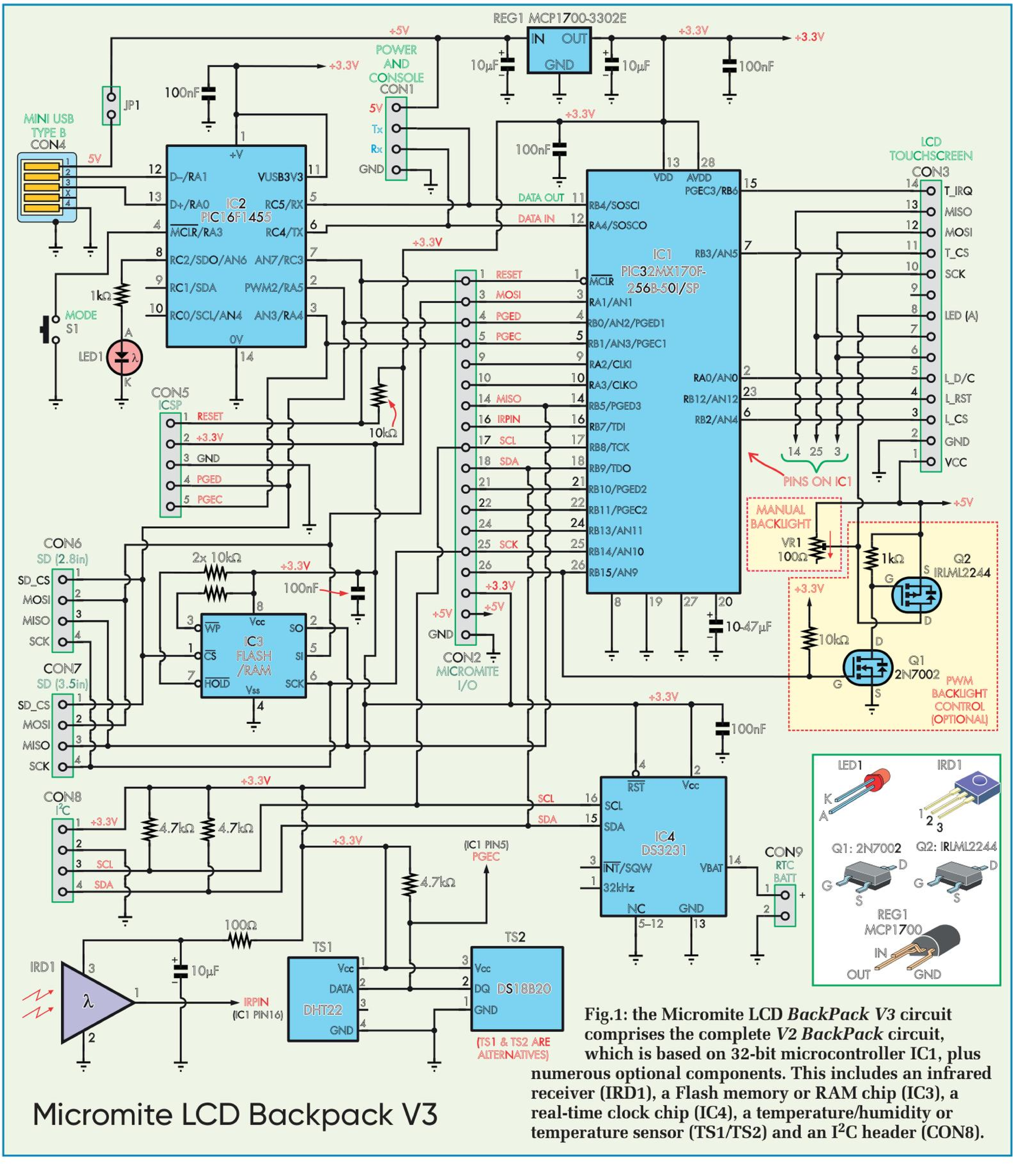

Micromite LCD Backpack V3
REG1 MCP1700-3302E
IN
OUT
GND
+5V
+3.3V
10μF
100nF
POWER AND CONSOLE
CON1
5V
Tx
Rx
GND
JP1
MINI USB TYPE B
CON4
IC2
PIC16F1455
D–/RA1
D+/RA0
MCLR/RA3
RC2/SDO/AN6
RC1/SDA
RC0/SCL/AN4
VUSB3V3
RC5/RX
RC4/TX
AN7/RC3
PWM2/RA5
AN3/RA4
+V
0V
1kΩ
MODE
S1
LED1
A
K
CON5
ICSP
RESET
PGED
PGEC
10kΩ
IC1
PIC32MX170F-256B-50I/SP
VDD
AVDD
PGEC3/RB6
DATA OUT
DATA IN
RB4/SOSCI
RA4/SOSCO
MCLR
RA1/AN1
RB0/AN2/PGED1
RB1/AN3/PGEC1
RA2/CLKI
RA3/CLKO
RB5/PGED3
RB7/TDI
RB8/TCK
RB9/TDO
RB10/PGED2
RB11/PGEC2
RB13/AN11
RB14/AN10
RB15/AN9
RB3/AN5
RA0/AN0
RB12/AN12
RB2/AN4
10-47μF
CON2
MICROMITE I/O
MOSI
MISO
IRPIN
SCL
SDA
SCK
LCD TOUCHSCREEN
CON3
T_IRQ
MISO
MOSI
T_CS
SCK
LED (A)
L_D/C
L_RST
L_CS
GND
VCC
PINS ON IC1
MANUAL BACKLIGHT
VR1 100Ω
1kΩ
Q2 IRLML2244
Q1 2N7002
PWM BACKLIGHT CONTROL (OPTIONAL)
CON6
SD (2.8in)
SD_CS
CON7
SD (3.5in)
2x 10kΩ
IC3
FLASH /RAM
WP
CS
HOLD
Vcc
Vss
SO
SI
CON8
I²C
4.7kΩ
IC4
DS3231
RST
INT/SQW
32kHz
NC
VBAT
CON9
RTC BATT
(IC1 PIN5) PGEC
100Ω
IRD1
IRPIN (IC1 PIN16)
TS1
DHT22
DATA
TS2
DS18B20
DQ
(TS1 & TS2 ARE ALTERNATIVES)
LED1
IRD1
Q1: 2N7002
Q2: IRLML2244
REG1 MCP1700
G
D
S
IN
OUT
GND

Fig.1: the Micromite LCD *BackPack V3* circuit comprises the complete *V2 BackPack* circuit, which is based on 32-bit microcontroller IC1, plus numerous optional components. This includes an infrared receiver (IRD1), a Flash memory or RAM chip (IC3), a real-time clock chip (IC4), a temperature/humidity or temperature sensor (TS1/TS2) and an I^{2}C header (CON8).

commonly available Arduino compatible I^{2}C modules, such as character LCD screens and other sorts of sensors.

4. DS3231 real-time clock IC

This (IC4) also uses the I^{2}C serial bus. It too has a 100nF bypass capacitor and a header (CON9) to connect a back-up battery so that the time and date are maintained even when the board has no external power. Micromite BASIC has built-in commands for I^{2}C-based real-time clocks, so this is another feature that can be used easily.

The I^{2}C pull-up resistors need to be installed if a DS3231 chip is installed, unless they are already present on another connected module.

5. Environment sensor

This is a DHT22 one-wire temperature and humidity sensor (TS1) or a DS18B20 one-wire digital temperature sensor (TS2). These connect to pin 5 of IC1, and there is provision for the required 4.7kΩ pull-up resistor too.

Data from the DHT22 can be read by a CFUNCTION which is available for download with the Micromite firmware, while there is a built-in BASIC function to read the temperature from a DS18B20. If fitting a DHT22, it's best to lay it over on its side over the top of the DS18B20 footprint to allow a display to fit above.

Software support

As noted above, we have written CFUNCTIONs to provide support for the 3.5-inch display; 2.8-inch and smaller displays based on the ILI9341 are natively supported by the Micromite.

The CFUNCTIONs for the 3.5-inch displays 'hook into' the existing graphics commands, so once the display has been initialised, the drawing commands are the same. If you have already written some MMBasic software, you only need to add a few lines at the start to support the higher-resolution 3.5-inch display.

The other great thing about our CFUNCTIONs is that they do not take complete control of the SPI bus, allowing other SPI devices to be used.

Unfortunately, these CFUNCTIONs interfere with the touch controller's BASIC functions, so we have had to write a separate set of CFUNCTIONs to handle the touch panel.

Most of the above-mentioned optional components are already supported by MMBasic, so we didn't need to write much code to allow you to use all the new features of the *V3 BackPack*. The only thing that is not natively supported is the Flash or SRAM memory IC, for which we've written demonstration code, as explained earlier.

High-value ceramic capacitors

Previous Micromites have required between one and three capacitors which were either specified as SMD 'chip' ceramics (10µF) or through-hole tantalum capacitors (47µF). This is because of the strict ESR requirements for some of the parts; 10µF tantalum capacitors often had too high an ESR to work reliably.

Some people didn't like having to solder the chip capacitors, and tantalum capacitors are more expensive and can be less reliable. Since then, through-hole 10µF ceramic capacitors have become available, and they use our preferred dielectric too (X7R). So we have specified those in the parts list. The other two options are still valid and can be used instead, if you prefer.

Construction

We'll start by assembling the basic *V3 BackPack* (effectively, a *V2 BackPack*), and then describe what parts to add if you want to use any of the optional features. See Fig.2, the PCB overlay diagram, to see which parts go where.

Begin by fitting the surface-mount components. This includes the mini-USB socket, CON4, and possibly some of the capacitors as well as MOSFETs Q1 and Q2 for PWM backlight control.

The pads for the mini-USB socket have been extended to make them easier to solder. Line the small posts in the underside of the socket up with the holes in the PCB; this should make everything else correspond. If so, solder one of the large mechanical pads in place to keep the socket in position and flush against the PCB.

Now apply flux to the pads for the electrical connections. You should be able to touch the iron to the pad extensions, allowing the solder to run up to the pins on the socket. Ensure the four pins are well attached and not bridged. If there are any bridges, carefully remove with solder wick. The pin with the shorter pad is not used in this application and does not need to be soldered.

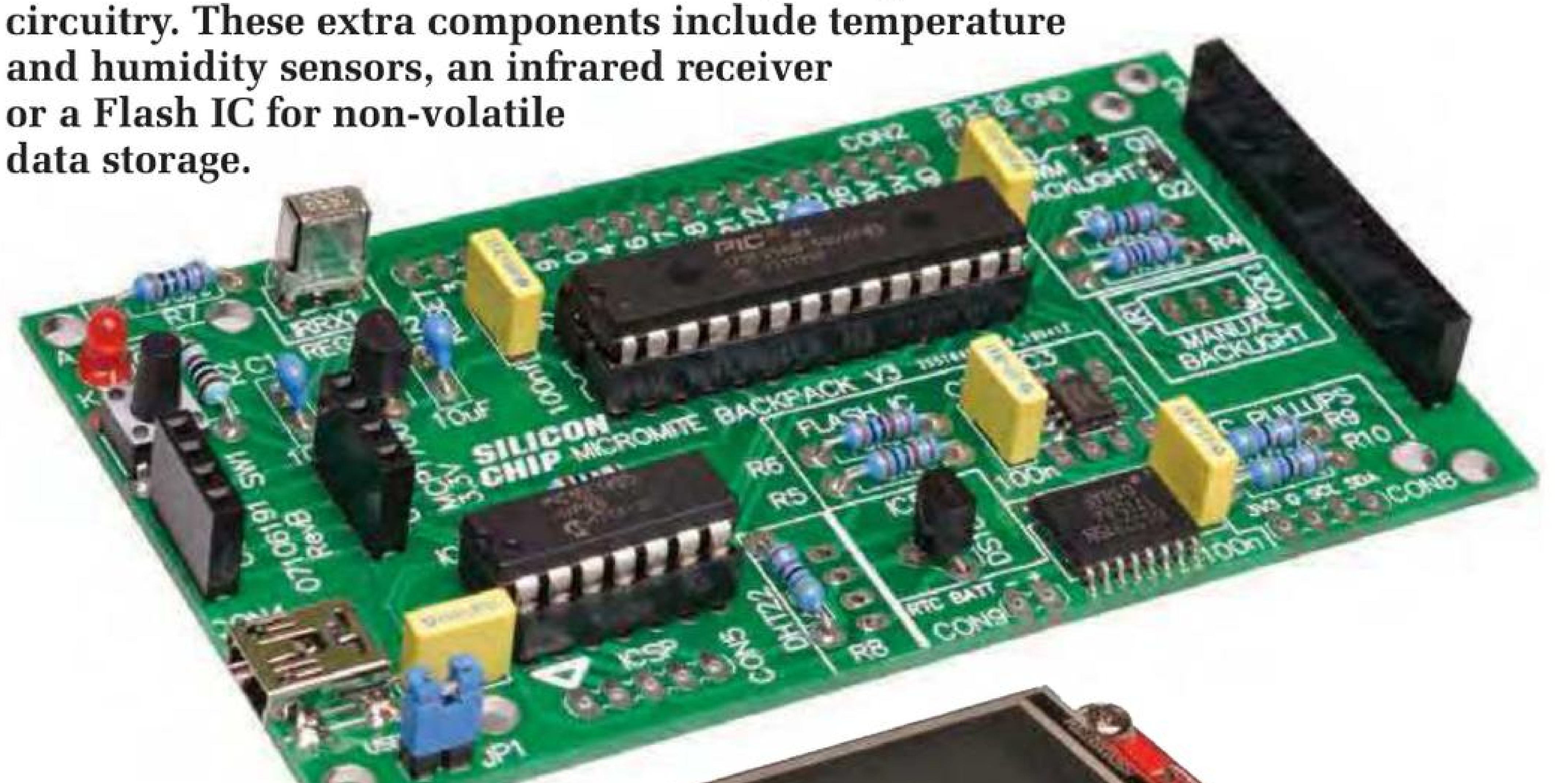
Solder the remaining mechanical pads to complete the attachment of the socket. Double check your work, as it will be difficult to access the pins later with the other components installed.

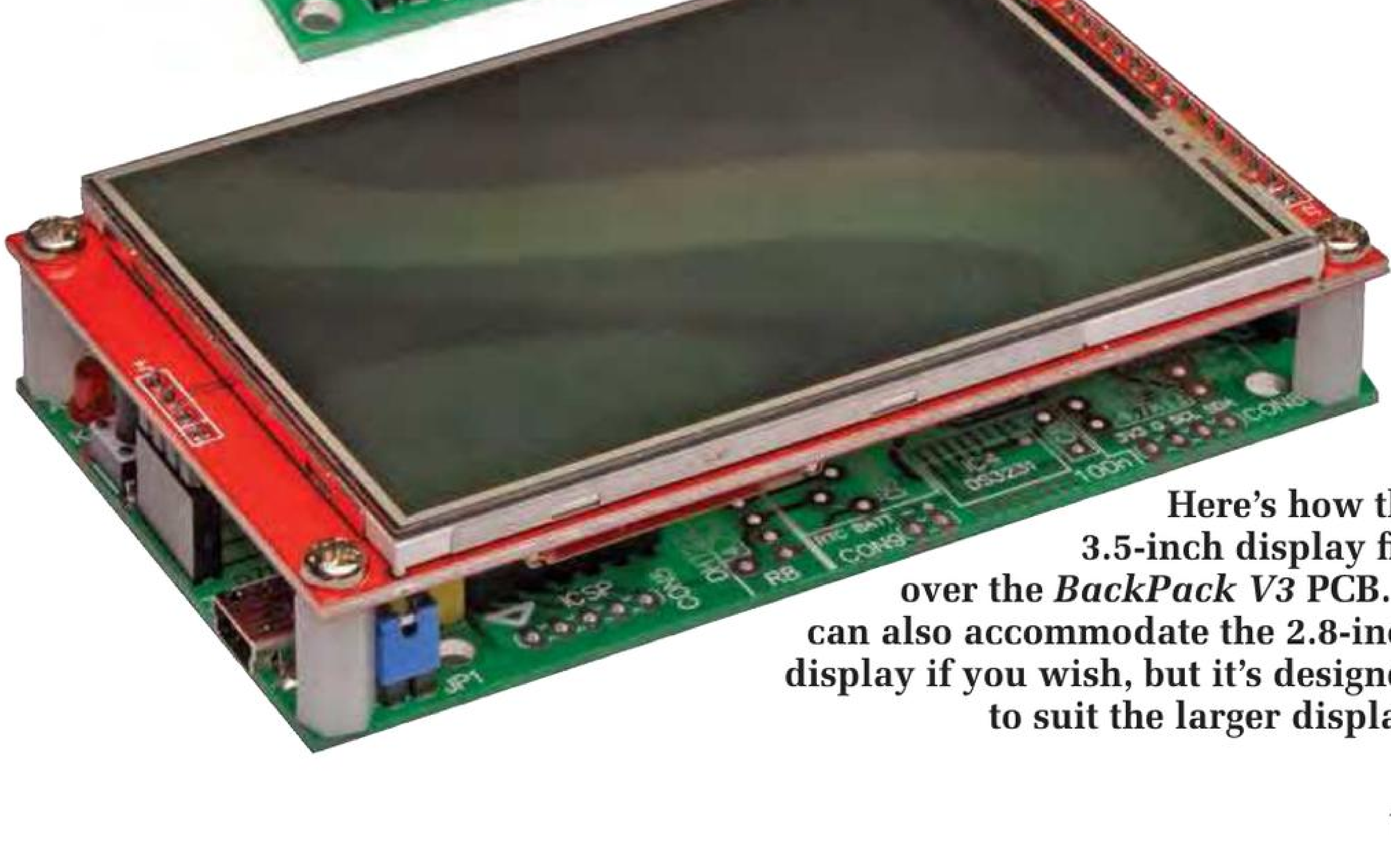
If you use SMD capacitors, they will all be the same type, but the two transistors are not. Check these are not mixed up before soldering them in place.

For the other SMD components, which are all quite small, an easy way to fit these is to apply solder to one of the pads then hold the component on top with tweezers. Apply the iron again to allow the solder to melt onto the component lead. This avoids having to handle three things at the same time.

If necessary, adjust the location of the parts so that the pins are fully lined up

The *V3 BackPack* can also be populated with other sensors and ICs to extend what it can do without requiring external circuitry. These extra components include temperature and humidity sensors, an infrared receiver or a Flash IC for non-volatile data storage.





Here's how the 3.5-inch display fits over the *BackPack V3* PCB. It can also accommodate the 2.8-inch display if you wish, but it's designed to suit the larger display.

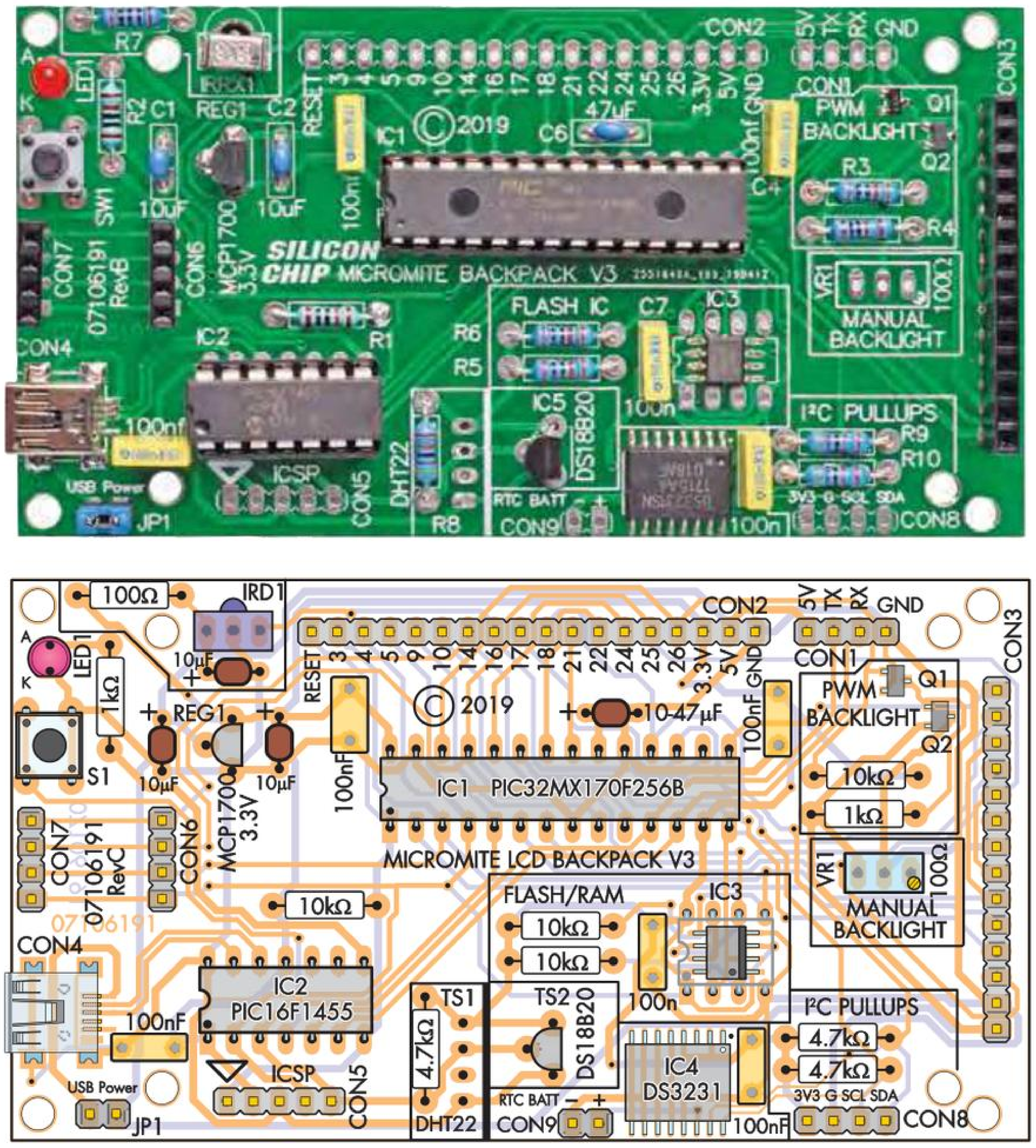


Fig.2: the slightly larger V3 *BackPack* PCB can accommodate a 2.8-inch (320×240 pixel) or 3.5-inch (480×320 pixel) LCD touchscreen, using the inner or outer set of four mounting holes respectively.

Both screens share the CON3 I/O and power connector, while CON6 makes electrical connections to the SD card socket on the smaller display, and CON7 on the larger display. CON2, the I/O header, is identical to that of the two earlier *BackPack* designs.

with the pads, and when you are happy, apply some solder to the remaining pins. Finally, go back and retouch the first pin to relieve any stress in the solder.

Through-hole parts

The remaining components can be added in the usual order. Fit the 10kΩ resistor between IC1 and IC2, and the 1kΩ resistor near LED1.

If you are using PWM backlight control, the two resistors below Q1 and Q2 must be fitted. (Remember to check resistor values with a DMM.)

Alternatively, you can fit potentiometer VR1 for manual backlight control, or a wire link as shown in our photo (below right) if you prefer to have the backlight fully on all the time.

If your potentiometer is more than 9mm tall, it may foul the display PCB and can be laid over in the space set aside if necessary.

Solder the capacitors next. If you are using tantalum capacitors, then the larger 47µF capacitor goes next to IC1, while the two 10µF capacitors sit either side of REG1. Tantalum capacitors are polarised, so take care that the positive leads go to the pads with a '+' sign.

If you are using ceramic capacitors instead, their polarity does not matter and you can use a 10µF ceramic in place of the 47µF tantalum, ie, all three high-value capacitors will be the same type.

There are three MKT or multi-layer ceramic through-hole capacitors to fit. Solder them in place and trim their leads.

Fit the two IC sockets next, if you are using them. These are a good idea in case you need to replace one of the chips. The notches on both face to the left, towards the USB socket. Note that if you do use sockets, IC2 will touch the underside of the SD card socket on the 3.5-inch display. This shouldn't cause any problems, but it can be avoided by separating the boards with 12mm tapped spacers.

The tactile switch sits near the left-hand edge of the board. Ensure it is pushed down firmly against the PCB before soldering its pins. It may take some force, but should pop into place. JP1 can also be fitted, below the USB socket. Unless you are powering the *BackPack* from an external 5V power supply, the jumper shunt will need to be fitted to source power from the USB socket.

The other headers should be fitted now. You will probably only need to install one of CON6 or CON7, depending on whether you are using a 2.8-inch or 3.5-inch display – do fit both if you wish to experiment.

It's a good idea to temporarily fit the headers onto the display you are using during soldering as this will keep the headers aligned squarely and correctly. CON3 can be fitted at the same time, to simplify lining up the display with the *BackPack*.

All that is left is to install the semiconductors. LED1 is mounted with its cathode (flat side) towards the USB socket. Ensure REG1's outline matches the footprint on the PCB and solder it down close to the PCB.

Fitting optional components

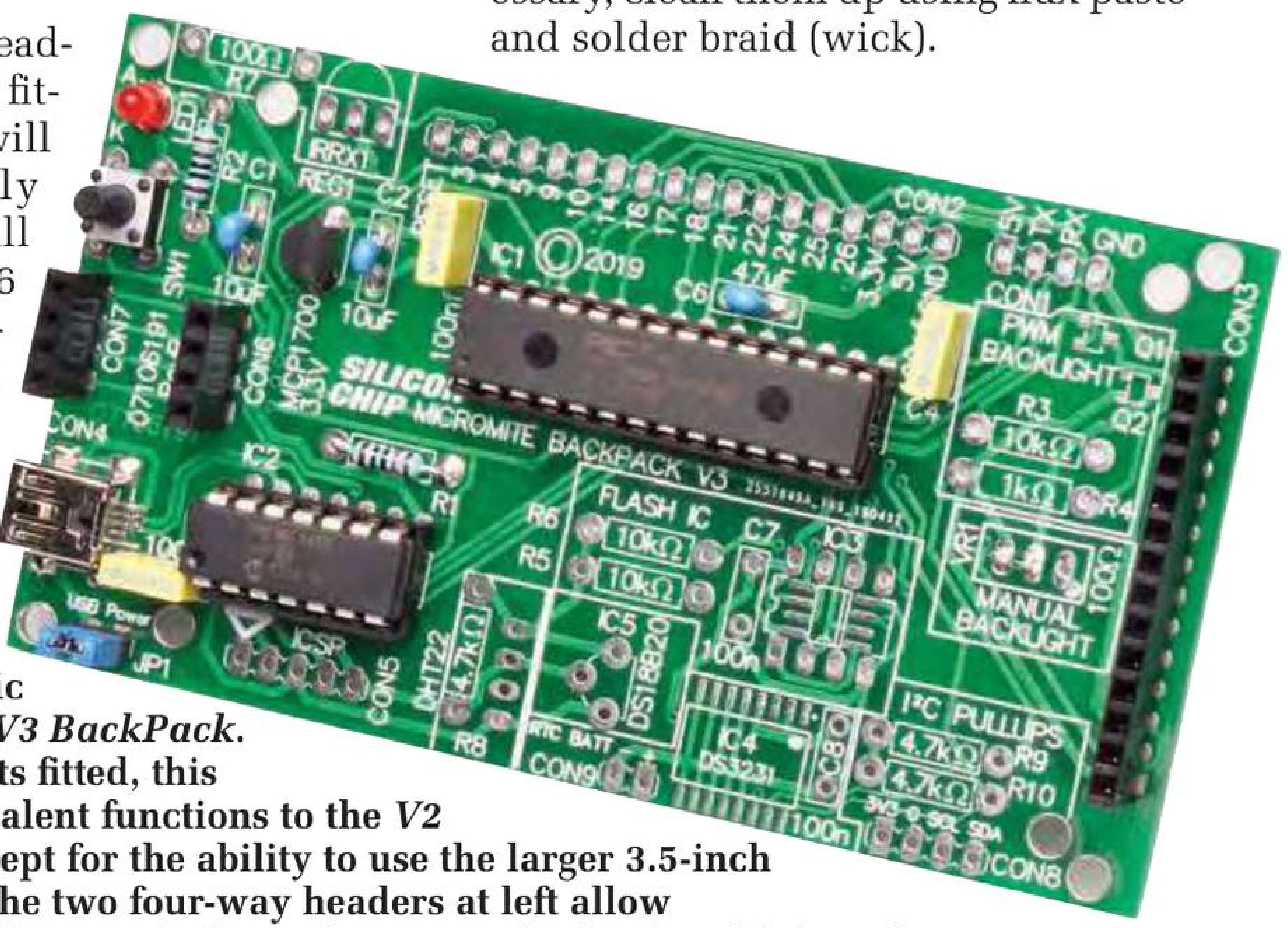
The parts list mentions what components you need to populate each optional add-on section.

These are all through-hole parts, except the Flash IC (IC3), which can be a through-hole or surface-mount type, and the DS32321 IC (IC4), which is only available in a surface-mount package.

If fitting IRD1, you also need to mount the adjacent 100Ω resistor and 10µF capacitor used to filter and bypass its supply.

It's a good idea to mount IRD1 with long enough leads that you can bend its lens to face in the same direction as the screen. It can be soldered on either side of the PCB, as long as its lead connections are not reversed compared to what is shown in Fig.2.

To fit IC4, the DS3231 IC, apply a small amount of flux to the pads and solder one pin in place. Check that its pin 1 dot is oriented as shown in Fig.2. Once you are happy that the part is flat and lined up with the other pins, carefully solder the rest. Ensure that no solder bridges have formed; if necessary, clean them up using flux paste and solder braid (wick).



This is the basic version of the *V3 BackPack*. With these parts fitted, this provides equivalent functions to the *V2 BackPack*, except for the ability to use the larger 3.5-inch touchscreen. The two four-way headers at left allow either a 2.8-inch or 3.5-inch touchscreen to be fitted to this board.

Parts list – MicroMite BackPack V3

(to provide the same functions as the V2 BackPack)

1 double-sided PCB, coded 07106191, 99 x 54.5mm, available from the PE PCB Service
1 mini USB type-B socket, SMD (CON4) [Altronics P1308]
1 SPST momentary tactile pushbutton (S1)
1 28-pin narrow (0.3in) DIL socket for IC1
1 14-pin DIL socket for IC2 (optional)
1 4-way header (CON1) (Micromite UART breakout; optional)
1 18-way straight header (CON2)
1 14-way female header (CON3)
1 5-way right-angle header (CON5) (for ICSP; optional)
1 4-way female header (CON6 or CON7)
1 2-way header and jumper shunt (JP1)
8 M3 x 6mm panhead machine screws (for mounting LCD)
4 M3 x 12mm tapped spacers (for mounting LCD)
1 2.8-inch or 3.5-inch LCD touch panel [eg, Silicon Chip Online Shop Cat SC3410]

Semiconductors
1 MCP1700-3302E/TO, TO-92 (REG1)
1 PIC32MX170F256B-50I/SP programmed with MMBasic firmware, narrow DIP-28 (IC1) – from **micromite.org**
1 PIC16F1455-I/P programmed with the Microbridge firmware, DIP-14 (IC2) – from **micromite.org**
1 3mm red LED (LED1)

Capacitors
3 10µF 16V X7R multi-layer ceramic capacitors (3216/1206 SMD or dipped leaded*) ***OR***
2 10µF 16V tantalum AND 1 47µF 16V tantalum
3 100nF 50V MKT polyester or multi-layer ceramic

Resistors (all 1/4W, 5%)
1 10kΩ 1 1kΩ

Optional parts for PWM backlight control
1 2N7002 N-channel MOSFET, SOT-23 (Q1)
1 IRLML2244TRPBF P-channel MOSFET, SOT-23 (Q2)
1 10kΩ 1/4W, 5% resistor
1 1kΩ 1/4W, 5% resistor

Optional parts for manual backlight control
1 100Ω 1/2W mini horizontal trimpot

Optional parts for infrared reception
1 three-pin 3.3V‡ infrared receiver (IRD1) [eg TSOP2136]
1 10µF 16V X7R multi-layer ceramic or tantalum capacitor (3216/1206 SMD or dipped leaded*)
1 100Ω 1/4W, 5% resistor *‡see text*

Optional parts for external RAM or Flash memory
1 SPI RAM or Flash IC, DIP-8 or SOIC-8 (IC3) [eg, 23LC1024 RAM or AT25SF041 Flash; pinout as in Fig.1]
1 100nF 50V MKT polyester or multi-layer ceramic capacitor
2 10kΩ 1/4W, 5% resistors

Optional parts for real-time clock
1 DS3231 RTC IC, SOIC-16 (IC4)
1 100nF 50V MKT polyester or multi-layer ceramic capacitor
2 4.7kΩ 1/4W, 5% resistors
1 2-pin header for CON9 (optional)
1 2.3-5.5V battery [eg, CR2032 lithium button cell; Jaycar Cat SB1762]

Optional parts for temperature/humidity sensor
1 DHT22 digital temperature and humidity sensor (TS1) ***OR***
1 DS18B20 digital temperature sensor, TO-92 (TS2)
1 4.7kΩ 1/4W, 5% resistor

Optional parts for external I^{2}C interface
1 4-pin header (CON8)
2 4.7kΩ 1/4W, 5% resistors ^

^ These resistors shared with RTC above.

Optional parts for temperature/pressure/altitude sensor
1 GY-68 BMP180 temperature/pressure sensor module ***OR***
1 GY-BMP280 temperature/pressure sensor module ***OR***
1 GY-BME280 temperature/pressure/humidity sensor
1 4-pin header (CON8)
2 4.7kΩ 1/4W, 5% resistors ^

* eg, Mouser Cat 810-FA26X7R1E106KRU6 or Digi-Key Cat 445-173437-1-ND

PCB, reduced kit and PCB with presoldered SMD parts
The *BackPack V3* PCB is available from the *PE PCB service*, A reduced kit of parts and PCB with the SMD parts pre-soldered are available – visit **micromite.org** for full details.

You will also need to fit the adjacent 100nF capacitor and both I^{2}C pull-up resistors (4.7kΩ). It's also a good idea to connect a battery (2.3-5.5V) via CON9. A CR2032 lithium battery is commonly used with the DS3231 and will last many years.

You can either solder its leads to the pads for CON9 or fit a pin header and connect the battery using patch leads or similar. If you don't connect a battery, IC4 will lose its time whenever power to the board is cut.

There isn't much room for a battery on the PCB (no mounting location is provided) so you'll have to figure out how to mount it (eg, double-sided tape) and wire it back to CON9. If mounting it somewhere on this PCB, make sure it's properly insulated so it can't short to any of the tracks or components.

Either the DHT22 (TS1) or DS18B20 (TS2) temperature sensor can be fitted, but not both. They connect to the same pin on the Micromite (pin 5) but use different communication protocols. They share a single 4.7kΩ pull-up resistor (inside the box labelled TS1), which needs to be fitted if either TS1 or TS2 is installed.

TS1 is tall, so it can be fitted laid over towards IC4 – the vented side of the case faces away from the PCB. If IC4 has already been fitted, there should still be room to lay TS1 on its side, but you will need to initially mount it slightly above the board so that it will sit flat on top of IC4 when bent over.

If fitting an SMD Flash or RAM chip for IC3, orient it with pin 1 towards the bottom edge of the board, as shown in Fig.2. You can solder it using a similar technique as described for IC4 above.

The through-hole version is easier to solder, and is oriented with its pin 1 dot or notch towards the left, as shown.

In either case, you will also need to fit the adjacent 100nF bypass capacitor and the two 10kΩ pull-up resistors. Note that some Flash ICs have internal pull-ups; in this case, you can omit those resistors. Check your device's data sheet to find out.

To connect an external I^{2}C module, including a BMP180 (GY-68 module), BMP280 (GY-BMP280 module) or BME280 (GY-BME280 module), fit pin header CON8 and the two 4.7kΩ resistors above it. As mentioned earlier, you can solder the module directly to CON8; match up its pinout, as printed on the module, with that shown in Fig.2 or on the PCB.

Note that some modules include pull-up resistors for the SDA and SCL lines. In this case, either don't fit the resistors on the *BackPack*, or remove them from the module. There should be just one set of pull-up resistors in the circuit.

Programming the chips

Both chips are available pre-programmed from our Micromite partner, **micromite.org**, but you only really need IC2 to be pre-programmed since it is capable of loading the software onto IC1, using **pic32prog** (see below). However, having IC1 pre-programmed will save you some effort.

While it is possible to program IC2 using a BASIC program on IC1 and a 9V battery, we only recommend this if you have no other way, and this has a bit of a 'chicken and egg' problem, in that it only works if IC1 has already been programmed. (See **http://geoffg.net/microbridge.html** for more information on this technique.)

You can program IC1 after fitting it, either using the ICSP header (CON5) and a PICkit or similar programmer, or by using IC2 in its *Microbridge* role. More information on using the *Microbridge* and its **pic32prog** software can be found in the article from May 2018 issue of *PE*.

We'll proceed assuming that you have pre-programmed chips, so fit them now. If you have used sockets, gently bend the leads of the ICs inwards to fit the sockets, otherwise, solder the chips directly to the PCB, taking great care that they are orientated correctly. Both ICs should have pin 1 facing towards the USB socket.

It's a good idea to solder two diagonally opposite corners and ensure the IC is flat and level before soldering the remainder.

The *V3 BackPack* is now usable and can be tested. Plug the *BackPack* into a computer and it should show up as a new USB-serial device.

Communication occurs at 38,400 baud on a freshly programmed Micromite, if you want to check this out now, using your favourite serial terminal program.

Drivers

Under Windows 10 and Linux, a driver should be automatically installed. If it is not, then the driver can be found at: **www.microchip.com/wwwproducts/en/MCP2200** – while this is a different device, it uses the same USB identification (VID and PID) codes as the *Microbridge* firmware.

(Incidentally, the MCP2200 is nothing more than a PIC18F14K50 that has been programmed to act as a USB-serial bridge, which is why this driver works).

When properly installed, the Micromite *BackPack* should appear as a new virtual COM port on your computer.

Configuring the display

The backlight controls work unchanged compared to the *V2 BackPack* (assuming you have fitted Q1, Q2 and their associated resistors).

The backlight intensity is set on a scale of 0 to 100 with the PWM function thus:

```
PWM 2,250,BACKLIGHT
```

This command works because pin 26 is the first output of PWM channel 2.

Alternatively, the backlight can be turned on or off by using the SETPIN and PIN commands to set the output of pin 26 high or low.

If you are using a 2.8-inch display, then the same instructions as given in the article from May 2018 (on the *V2 BackPack*) apply. The following commands initialise and calibrate the display:

```
OPTION LCDPANEL
  ILI9341,L,2,23,6
GUI TEST LCDPANEL
OPTION TOUCH 7,15
GUI CALIBRATE
GUI TEST TOUCH
```

The 3.5-inch panel works slightly differently, as it depends on a CFUNCTION to work and is not quite as 'transparent' as the inbuilt display driver.

See the panel titled *Driving the 3.5-inch touchscreen* for details on how to set up and use the larger screen.

If you have fitted any of the optional components, see the opposite panel, *Using the optional components*, which describes the software required to use them.

Reproduced by arrangement with
SILICON CHIP magazine 2020.
www.siliconchip.com.au

Breaking news from The Back Shed

While we were preparing this article, Geoff Graham told us that Peter Mather had made a post on his forum, 'The Back Shed', describing a driver that he had created for the ILI9488 display controller.

The Back Shed is a great place to get information on the various Micromites and other topics; see: **www.thebackshed.com/forum/**

His code for the display controller can be found at: **www.thebackshed.com/forum/forum_posts.asp?TID=11419**

It is implemented as a CSUB which is run by the Micromite at startup. The initialisation process is different to our CFUNCTION, but after that, you use the same native graphics commands as with our code. The code shown on the forum is for a different Micromite board, so the initialisation line needs to be changed to suit the pinouts used on the *BackPack*. Copy and paste his code labelled 'MM2' into a blank program, then change the second line from:

```
ILI9488 16,2,9,1
```

to:

```
ILI9488 2,23,6,1
```

These parameters determine the display CD pin, RST pin, $\overline{\text{CS}}$ pin and orientation. This changes the pin values to suit the *BackPack*. The orientation is a value from 1 to 4, as explained in the main text of this article. Upload the program to the Micromite and run the command:

```
LIBRARY SAVE
```

To store the CSUB as a library instead of BASIC code, restart the processor with the command:

```
WATCHDOG 1
```

The driver will then be loaded. At this stage, the Micromite is at the same state as if the OPTION LCDPANEL command had been run for the 2.8-inch screen, and normal touch panel initialisation can continue, like this:

```
GUI TEST LCDPANEL
OPTION TOUCH 7,15
GUI CALIBRATE
GUI TEST TOUCH
```

Readers who are comfortable with the usual way of setting up touch panels on the Micromite, such as the ILI9341, may prefer this method as it works similarly. However, note that you will lose the ability to use the SPI peripheral for other purposes, as is the case with the 2.8-inch display.

Peter also noted the glitch with the MISO pin on these displays which we found (and worked around) while trying them out in our June 2020 3.5-inch displays article and then on the *V3 BackPack* board.

Finally, future releases of the Micromite V2 firmware will include a copy of Peter Mather's ILI9488 CSUB driver.

Using the optional components

Using the infrared receiver (IRD1)

MMBasic only supports an infrared receiver on pin 16 of the 28-pin PIC32, so that is where we have connected it. You therefore lose this pin as a general purpose I/O when you fit IRD1.

MMBasic can trigger a software interrupt when a valid command is received and then call a user-defined subroutine; set up as follows:

```
IR DevCode, KeyCode, IR_Int
```

DevCode and KeyCode specify the variable names which will contain the device and key codes respectively when the user routine ('IR_Int' in this case) is called. So you could define the function like this:

```
SUB IR_Int
 PRINT  "DEVICE:" DevCode "KEY:"
  KeyCode
END SUB
```

Using the real-time clock

MMBasic has built-in routines to use an RTC module connected to the hardware I^{2}C pins, as is the case here. Set the Micromite's internal clock from the DS3231 IC thus:

```
RTC GETTIME
```

Setting the time on the DS3231 is done with a single command specifying the current date and time:

```
RTC SETTIME year, month, day, hour,
  minute, second
```

If you are using any other I^{2}C devices, you can connect them via CON8. If, as is often the case, the module(s) have their own pull-up resistors, either remove them or omit the onboard I^{2}C pull-up resistors. It may work with both in place, but this is not recommended

Temperature and humidity sensors

The temperature from a DS18B20 (TS2) can be read with a single MMBasic command:

```
TEMPERATURE = TEMPR(5)
```

Functions for communicating with a DHT22 were built into early versions of MMBasic, but have been removed in later versions; instead, a CSUB is supplied to do the same job. The required code and documentation can be found in the 'Humid.pdf' file in the 'Embedded C Modules' subfolder of the Micromite firmware download, available from **http://geoffg.net/micromite.html#Downloads**

After the CSUB has been copied into the BASIC program, the temperature and humidity can be read by a single command like this:

```
HUMID 5, TEMPERATURE, HUMIDITY
```

The first parameter (5) tells this function which Micromite pin the DHT22 sensor is connected to. The results are saved in the TEMPERATURE and HUMIDITY variables. Due to the way the DHT22 works, the results are actually from the previous time this command was issued, with the current call starting the next conversion in the background.

Therefore, you will need to ignore the values of TEMPERATURE and HUMIDITY the first time you call this command. Hence, it's a good idea to issue it during your initialisation routine.

Using a RAM chip

We test-fitted our board with a 23LC1024 RAM IC (IC3). It's similar to the 23LCV1024 used in the *433MHz Wireless Range Extender* project (see *PE*, May 2020).

There is no WP (write-protect) function on the RAM IC, but it does have a HOLD pin which needs to be held high, so the 10kΩ pull-up resistors are still required.

We've written a sample program to demonstrate using such a chip, which is named '23LC1024 RAM IC.bas'. It simply writes data to the chip, based on the contents of the TIMER variable, then reads those values back and prints them out on the Micromite terminal.

The CS pin of IC3 is hardwired to the Micromite's pin 4, and this is set as a constant at the start of the sample program. The SETRAMMODE subroutine provides page, byte and sequential options. Using the sequential option means that the entire RAM contents can be read or written in one pass.

A read or write starts with a STARTRAMREAD/STARTRAMWRITE command, which pulls CS low and sends a command sequence containing the supplied start address.

After that, subsequent calls to RAMREAD or RAMWRITE read or write a single byte before incrementing the address pointer. The sequence ends with a call to ENDRAMREAD/ENDRAMWRITE which brings CS high, releasing the SPI bus.

Using external Flash memory

For testing out the Flash interface, we tried an AT25SF041 4Mbit (0.5MB) Flash IC (again, as IC3). On this chip, the $\overline{\text{WP}}$ and $\overline{\text{HOLD}}$ pins are internally pulled high, so the 10kΩ resistors are not needed, although they were fitted to our prototype; it doesn't hurt to have both internal and external pull-ups.

Writing to the device is a bit more complicated than for a RAM chip, but reading uses the same command and format as the RAM IC.

Flash memory cannot usually be written byte by byte. An entire 'page', 4KB in this case, must be erased (set to all 1s), then data can be written byte by byte (or 'programmed' according to the data sheet terminology). Writes occur in blocks of up to 256 bytes. The data to be written is actually stored into a RAM buffer; it isn't written to Flash until the CS line goes high, at the end of the process.

There are a few more details than what's described here; so the device data sheet is a good place to check out the subtleties of the process. One wrinkle, for example, is that the writes will wrap around at addresses that are multiples of 256 bytes. There is also a software flag (WEL; write-enable latch) that must be set before any changes (erase or write) can occur to the Flash memory contents. Thus, a typical write sequence would consist of setting the WEL flag, erasing a page, setting the WEL flag again and then writing the actual data.

The sample program is called 'AT25SF041 FLASH IC.bas'. Unlike the RAM demo, which loops continuously, this program reads the Flash once, writes data to the Flash once, then rereads it, displaying the results on the terminal. This is to avoid wearing out the Flash.

The Flash chip we used has a minimum endurance of 100,000 cycles, which means that it would take 27 hours at one write per second (to the same part of the Flash memory) to potentially cause a failure.

Using a BMP180, BMP280 or BME280 sensor module

We published an article in the December 2018 issue of *PE* explaining how to use a BMP180 or BMP280 module with a Micromite. You can download the sample BASIC code for free from the August 2020 page of the *PE* website.

The BMP180/BMP280 provide temperature and pressure/altitude data, but the BME280 also includes humidity. You can find MMBasic source code to read data from a BME280 sensor at the (very useful) BackShed forum: **www.thebackshed.com/forum/forum_posts.asp?TID=8362**

Driving the 3.5-inch touchscreen

When using the 2.8-inch touchscreen, you set it up once using the OPTION command (as described in the main text) and from then on, the Micromite automatically configures it each time the chip is powered up. But because MMBasic doesn't natively support the 3.5-inch touchscreen, setting it up is a bit different.

You need to run some code at the start of your program, every time the chip is powered up, to configure this display. This code initialises the display and also sets up the 'hooks' into Micromite BASIC's graphics commands so that you can draw to this screen using the same commands as for the 2.8-inch display.

One big difference of this implementation is that it does not block use of the SPI pins to other interfaces. In fact, the user program must start the SPI peripheral just as for any other interface. This is also the reason why the in-built touch commands won't work, as they too require exclusive use of the SPI interface.

Although the control pins for the LCD and touch controllers (such as $\overline{\text{CS}}$, DC and RESET) are hardwired into the CFUNCTION to match the hardware that is on the *BackPack*, they need to be set up by the user program. The advantage here is that control can be taken back if your program wants to use these pins for other purposes.

The CFUNCTION assumes that all this setting-up has been done, and will fail if it has not. This is so that the CFUNCTION has minimal overhead and is thus quite fast. This is handy, as the 3.5-inch displays have twice as many pixels to manage as the 2.8-inch displays.

The following code needs to appear before the display functions can be used with the 3.5-inch display. You can also find this code in our example programs:

```
DIM INTEGER ROTATION=1,BUCKET,
  ILI9488_SPI_ADD
ILI9488_SPI_ADD=PEEK(CFUNADDR ILI9488_SPI)
SPI OPEN 20000000,0,8
SETPIN 2,DOUT
SETPIN 23,DOUT
SETPIN 6,DOUT
BUCKET = ILI9488_SPI(ILI9488_SPI_ADD,
  ROTATION)
```

The first line defines three integer variables. ROTATION sets the display orientation. Set it to a value between one and four. Mode one is portrait, two is landscape, three is upside-down portrait and four is upside-down landscape.

BUCKET (the 'bit-bucket') is used as a place to store the return value of the CFUNCTION. BASIC insists on us storing the return value of a function when calling it, so even though we don't need to use that return value, we need somewhere to store it.

ILI9488_SPI_ADD is used to hold the Flash memory address (shortened to 'ADD') of the CFUNCTION. This needs to be passed to the CFUNCTION during the initialisation stage, as it needs this to set up the hooks into the native graphics functions.

The address of the CFUNCTION is retrieved by using the PEEK function on the second line. We have called the CFUNCTION 'ILI9488_SPI', so if you change this, you will need to change that second line too.

The next four lines set up the micro's $\overline{\text{SPI}}$ peripheral and set up the I/O pins used to control the screen's $\overline{\text{CS}}$, DC and RESET lines.

Finally, the display is initialised by our CFUNCTION according to the ROTATION setting. After this, you will normally clear the screen using a command like this:

```
CLS(RGB(BLACK))
```

Our demonstration program, 'ILI9488_SPI_minimal working. bas', can be downloaded from the August 2020 page of the *PE* website. This sets up the display as described above and then draws some patterns on the screen using the inbuilt graphics functions.

Using the touch interface

As mentioned in the text, MMBasic's built-in touch panel support doesn't play well with our new driver. We suspect that this is because the display driver is not initialised when the touch controller attempts to start up at Micromite boot time. So we have written a separate CFUNCTION to provide the touch functions.

The 'ILI9488 with touch calibration.bas' demonstration program (in the same download package from the *PE* website) shows how to read raw touch data and also calculate touch locations on the screen. In addition to initialising the display controller, as noted above, the following lines are required to use the touch controller:

```
DIM INTEGER TOUCH_X0,TOUCH_Y0,
  TOUCH_X1,TOUCH_Y1
TOUCH_X0=110
TOUCH_Y0=1993
TOUCH_X1=2001
TOUCH_Y1=76
SETPIN 7,DOUT
```

These four variables provide touch panel calibration. Our calibration sketch generates a new set of calibration values for a specific touch panel, which can be pasted back into your program.

The ROTATION variable also needs to be set, as described earlier, since the calibrated touch coordinates depend on the display rotation that is being used.

The last line sets up the Micromite pin used to drive the touch controller's $\overline{\text{CS}}$ (chip select) line.

To retrieve the x-axis component of the current touch position, use the following CFUNCTION call:

```
X=XPT2046(0,ROTATION,TOUCH_X0,TOUCH_Y0,
  TOUCH_X1,TOUCH_Y1,MM.HRES,MM.VRES)
```

This CFUNCTION requires no initialisation, although it assumes that the SPI interface has already been set up, as this is required to use the display anyway. This CFUNCTION reduces the speed of the SPI bus below the 2.5MHz limit of the touch controller IC for the duration of the CFUNCTION, and returns it to its previous value afterwards.

To read the y-axis, the value of one is passed as the first parameter instead:

```
Y=XPT2046(1,ROTATION,TOUCH_X0,TOUCH_Y0,
  TOUCH_X1,TOUCH_Y1,MM.HRES,MM.VRES)
```

To retrieve the raw ADC values (which are necessary for the calibration), values of two, three or four are passed as the CFUNCTION's first parameter. The z-axis value (with the first parameter as four) corresponds to the pressure on the touch panel, and is used by our function to check whether a valid touch is occurring; for example:

```
RAWX=XPT2046(2)
RAWY=XPT2046(3)
RAWZ=XPT2046(4)
```

By using the z-axis value, the IRQ pin on the touch controller is not needed for the 3.5-inch displays, although it is left connected on our board, for use with the 2.8-inch displays.

STEERING WHEEL AUDIO BUTTON TO INFRARED ADAPTOR

by John Clarke

SILICON CHIP
Steering Wheel Control Adaptor

ainment' head
b steering wheel controls, those
. 's because many aftermarket head
units do not support steering wheel controls, the implementation of which often varies between manufacturers and even between models. This new adaptor lets you use most of those very handy controls with a wide range of aftermarket head units.

Once upon a time (would you believe way back in 1930?) car manufacturers started fitting car radios. Nothing fancy, mind you – just a basic AM receiver.

Over the years, buyers demanded more: push-button tuning, FM tuners, 8-track players, cassette players, CD/DVD players and so on. In more recent times, we've seen that expand to include auxiliary inputs, USB and SD-card readers, Bluetooth and of course inbuilt navigation systems.

To control all this technology, 'head units' were created – essentially a dedicated computer in its own right – with not just the source but such things as volume, radio station, track selection and more selected via push-buttons and, becoming more popular, an infrared remote control.

Then someone tried incorporating those push-buttons into the steering wheel – and the Steering Wheel Controller (SWC) was born, offering remote control without taking your eyes off the road for very long (if at all).

Some head units incorporate a remote control input wire at the rear of the unit and are operated via a voltage or digital signal.

Fortunately, with our *Adaptor* it doesn't matter which system the head unit supports (if any) – just so long as it also offers infrared remote control. Almost all modern head units do.

These handheld remotes are small and fiddly to use, and we don't recommend that they're used by the driver because they are too distracting. That's if the driver can find it in the first place: they have an annoying habit of falling down between the seats.

Our *SWC Adaptor* can operate the head unit using infrared control and it is, in turn, controlled by the steering wheel buttons. So you don't even need to open up your head unit to use it. You can feed the IR control signals in through the faceplate.

Note that some SWCs are digital; they may be connected via a Controller Area Network (CAN) bus or a proprietary system. These are not suitable for use with this *Adaptor*. It works with controls where each switch connects a different resistance between a particular wire and chassis (0V) when pressed.

Before embarking on this project, it would be wise to check that your steering wheel controls are suitable for use with our *SWC Adaptor*. See the panel titled, *Are your steering wheel controls suitable?*

The only other requirement is that the head unit uses one of these three infrared remote control protocols: NEC, Sony or Philips RC5. Virtually all head units with remote control use one of those three.

By far the most common is the NEC format, used by most head units manufactured in Asia, including Pioneer, Akai, Hitachi, Kenwood, Teac and Yamaha, plus Blaupunkt in Germany.

The Sony protocol is the next most common. The RC5 format is used by Philips and some other European brands, although we have seen some Philips products using the Sony format

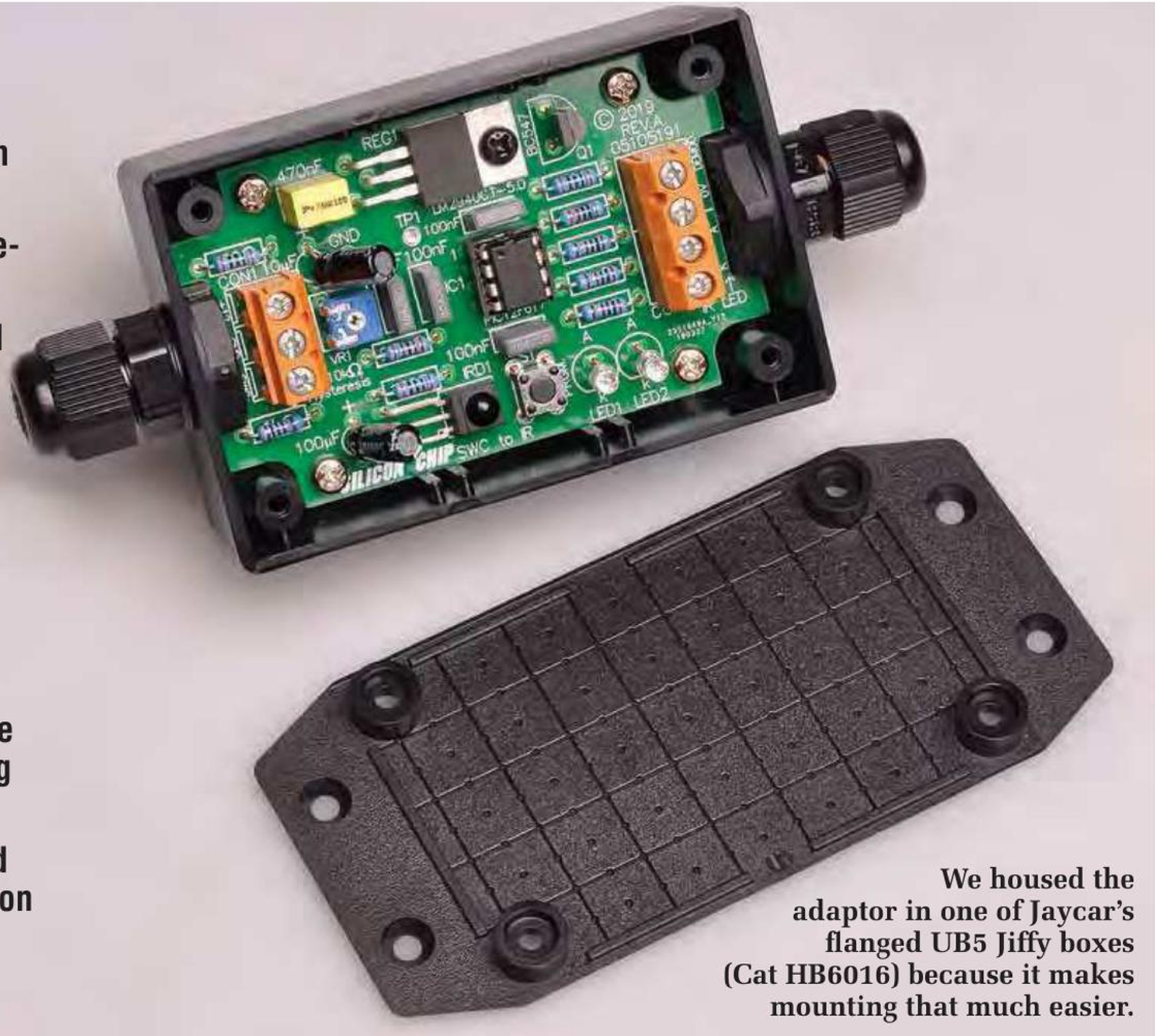
Presentation

The *SWC Adaptor* comprises a small PCB which can fit into a small Jiffy box. It's connected to an ignition-switched 12V supply and the steering wheel control wire. It provides two outputs: one to drive an infrared LED to operate the head unit, and a second for an optional direct wire connection which can control the head unit directly, without the need for an infrared transmitter – more on that later.

In use, the *SWC Adaptor* can be programmed to map up to ten steering wheel buttons to separate infrared codes to send to the head unit. Once programmed, it can be hidden away (eg, under or behind the dash) and the steering wheel buttons can be used to control the head unit while the vehicle ignition is on.

Features

- Compact unit, can be hidden away under or behind the dash or even inside the head unit
- Works with up to 10 resistance-based steering wheel buttons
- Controls head unit via infrared signals (requires remote control capability)
- Works with most head units (using NEC, Sony or RC5 infrared codes)
- Infrared receiver included for programming the function of each button
- Easy set-up by learning remote control codes for each steering wheel button
- Optional unmodulated infrared output for direct wire connection
- Two non-repeat buttons for special functions (see text)



We housed the adaptor in one of Jaycar's flanged UB5 Jiffy boxes (Cat HB6016) because it makes mounting that much easier.

Circuit description

Fig.1 shows the circuit of the *SWC Adaptor*. It is based around microcontroller IC1, a PIC12F617-I/P. This monitors the steering wheel controls via analogue input AN3, while also sensing tolerance adjustment trimpot (VR1) at analogue input AN1, the state of switch S1 at digital input GP5 and the signal from infrared receiver IRD1 at digital input GP3.

To control the vehicle head unit, IC1 produces remote control code pulses at its pin 5 PWM output. These codes are transmitted in 36-40kHz bursts, to drive infrared LED3. An identical, non-modulated signal is also sent to the GP0 digital output (pin 7).

This has the advantage that you can wire it in place of the infrared receiver, for a direct wired connection to the head unit. The exact modulation frequency depends on the infrared protocol that the unit is set up for. It is 36kHz for the Philips RC5 protocol, 38kHz for the NEC protocol and 40kHz for the Sony protocol.

In more detail, the SWC input at CON1 has a 1kΩ pull-up resistor to the 5V supply. This forms a voltage divider across the 5V supply, in combination with the steering wheel switch resistances, giving a different voltage at analogue input AN3 (pin 3) of IC1 for each switch that is pressed.

This voltage is applied to the AN3 input via a low-pass filter comprising a 2.2kΩ resistor and 100nF capacitor. IC1 converts the 0-5V voltage to a digital value between 0 and 255.

So for example, a 2.5V signal would be converted to a value of 127 or 128, around half of the maximum value of 255.

As for the AN1 input, the 0-5V from trimpot VR1's wiper is converted to a digital value. The 0-5V range of VR1 is mapped in software to a 0-500mV range of tolerance.

If VR1 is set midway at 2.5V, the tolerance setting is 250mV (1/10th of the wiper voltage, measured at TP1). So the SWC input voltage can differ from its stored value by up to ±250mV and still be recognised as that particular switch.

Tolerance is essential since the SWC voltage may vary with temperature due to resistance variation in the switch resistor; switch contact resistance can also cause voltage variation.

Are your steering wheel controls suitable?

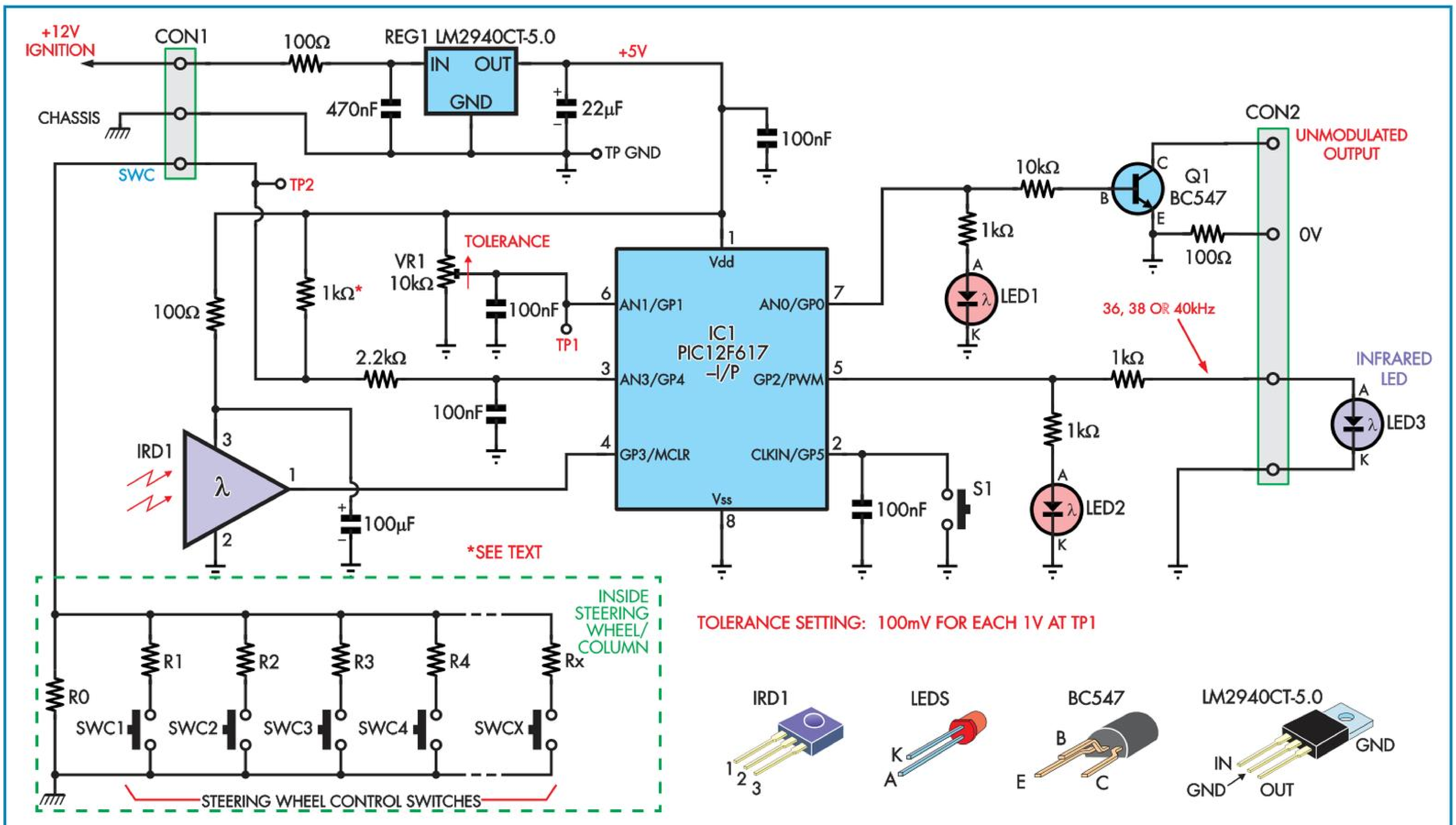
Before deciding to build the *SWC Adaptor*, you will need to check that the steering wheel control switches are the type that switch in a resistance rather than digital types that produce a series of digital (on and off) signals when the switch is pressed. We also assume that the head unit you intend to use has infrared remote control and uses one of the standard protocols mentioned in the article.

To check the SWC switches, your original equipment head unit will offer clues as to which wire this is. There should be a connection diagram on the head unit. Or you can find the wire using a vehicle wiring diagram.

With the ignition off and the SWC wire not connected to the head unit, connect your multimeter leads between that wire and vehicle chassis. Set the multimeter to read resistance. The resistance may read very high ohms when the SWC switches are all open or it may be a few thousand ohms. Pressing each SWC switch in turn should show a different resistance reading.

For example, our test vehicle showed a resistance of 3.5kΩ with all switches open. Then the switch readings were 160Ω, 79Ω, 280Ω, 450Ω, 778Ω and 1.46kΩ for each of the six switches. So these readings prove that the steering wheel controls are the analogue type that switch in resistance and so is suitable for use with the *SWC Adaptor*.

If you do not get resistance changes, check that you are monitoring the correct wire and that the chassis connection is good. If the switches still do not show resistance, they might be producing a digital signal when the vehicle ignition is on. The steering wheel controls on your vehicle are therefore not suitable for use with the *SWC Adaptor*.


+12V
IGNITION
CON1
100Ω
REG1 LM2940CT-5.0
IN
OUT
GND
+5V
CHASSIS
470nF
22µF
TP GND
100nF
SWC
TP2
CON2
UNMODULATED
OUTPUT
10kΩ
C
B
E
Q1
BC547
1kΩ
100Ω
0V
TOLERANCE
VR1
10kΩ
1kΩ*
100Ω
100nF
TP1
1
Vdd
6
AN1/GP1
AN0/GP0
7
IC1
PIC12F617
–I/P
A
LED1
K
36, 38 OR 40kHz
2.2kΩ
3
AN3/GP4
GP2/PWM
5
1kΩ
INFRARED
LED
100nF
1kΩ
A
LED3
K
IRD1
3
4
GP3/MCLR
CLKIN/GP5
2
1
100nF
S1
A
LED2
K
100µF
Vss
8
2
*SEE TEXT
INSIDE
STEERING
WHEEL/
COLUMN
TOLERANCE SETTING: 100mV FOR EACH 1V AT TP1
R1
R2
R3
R4
Rx
R0
SWC1
SWC2
SWC3
SWC4
SWCX
STEERING WHEEL CONTROL SWITCHES
IRD1
1
2
3
LEDS
K
A
BC547
B
E
C
LM2940CT-5.0
GND
IN
GND
OUT

Steering Wheel Control Adaptor

Fig.1: IC1 monitors the steering wheel controls via analogue input AN3, while also sensing the tolerance adjustment trimpot (VR1) at analogue input AN1. The state of switch S1 is monitored at digital input GP5, and the signal from infrared receiver IRD1is monitored at digital input GP3. To control the vehicle head unit, IC1 produces remote control code pulses at its pin 5 PWM output. These codes are transmitted in 36-40kHz bursts, to drive infrared LED3. An identical, non-modulated signal is also sent to the GP0 digital output (pin 7). This has the advantage that you can wire it in place of the infrared receiver, for a direct-wired connection to the head unit.

Having detected a valid SWC button press, IC1 activates its pin 5 and 7 outputs to produce the appropriate remote control code to send to the vehicle head unit.

The modulated output at pin 5 has a 50% duty cycle. It can drive an infrared LED via a 1kΩ resistor and CON2. LED2 is also driven by the PWM output during transmissions – a visible indicator.

The unmodulated output from pin 7 drives the base of NPN transistor Q1 via a 10kΩ resistor and also LED1, via a 1kΩ resistor. The collector of Q1 is open so that it can connect directly to the IR receiver in the head unit. The emitter is isolated from ground via a 100Ω resistor to reduce current flow due to the possibly differing ground potentials in this unit and the head unit.

Fig.2 shows the output signals at pins 5 (yellow) and the collector of Q1 (cyan), demonstrating the 36-40kHz modulation applied to pin 5 but not Q1's collector. In this case, the NEC protocol is being used so the modulation is at 38kHz.

The unit is set up using infrared receiver IRD1. This three-pin device incorporates an infrared photodiode, amplifier and automatic gain control plus a 38kHz bandpass filter to accept only remote control signals, within a few kHz of the carrier frequency.

The filter is not narrow enough to reject the 36-40kHz frequencies that could be produced by various different remote control units.

IRD1 removes the carrier, and the resulting digital signal is fed to the GP3 digital input of IC1 (pin 4), ready for code detection.

IRD1 runs from a 5V supply filtered by a 100Ω resistor and 100µF capacitor, to prevent supply noise causing false IR code detection.

Pushbutton switch S1 is bypassed with a 100nF capacitor to filter transients and for switch debouncing. The voltage at digital input GP5 is held at 5V via a weak pull-up current, internal to IC1.

When S1 is pressed, GP5 is pulled low to 0V and IC1 detects this. S1 is used during programming and to set a new tolerance adjustment.

The circuit is powered from the vehicle's 12V ignition-switched supply, fed in via CON1. This supply goes through an RC low-pass filter (100Ω/470nF) and then to automotive 5V linear regulator REG1, to power IC1 and the rest of the circuitry.

The LM2940CT-5.0 regulator will not be damaged with a reverse supply connection or transient input voltage up to 55V, for less than 1ms. These situations can occur with some regularity in vehicle supplies; for example, with an accidentally reversed battery or when windscreen wiper motors switch off.

Construction

The *SWC Adaptor* is built on a PCB coded 05105191, measuring 77 × 47mm, available from the *PE PCB Service*. It fits into a UB5 Jiffy box. The overlay diagram (Fig.3) shows how the components are fitted.

Start with the resistors – use a multimeter to check the value of each set of resistors before fitting them, as colour codes can be confusing.

We recommend using a socket for IC1. Take care with the orientation when installing the socket and IC1.

The capacitors can be fitted next. The electrolytic types must be installed with the polarity shown, with the longer positive lead towards the top of the PCB. The polyester capacitors (MKT) can be mounted with either orientation on the PCB.

REG1 is installed next. It's mounted horizontally on the PCB. Bend the leads so they fit the PCB holes with the tab mounting holes lining up. Secure the regulator to the PCB with the screw and nut before soldering the leads.

Infrared receiver IRD1 also mounts horizontally, with the lens facing up and the leads bent 90° to fit the holes.

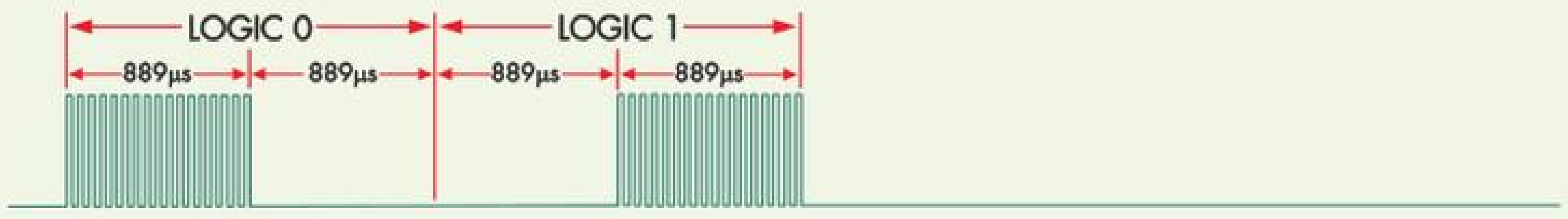
Infrared coding

Most infrared controllers switch their LED on and off at a modulation frequency of 36-40kHz in bursts (pulses), with the length of and space between each (pauses) indicating which button was pressed. The series of bursts and pauses is in a specific format (or protocol) and there are several commonly used. This includes the Manchester-encoded RC5 protocol originated by Philips.

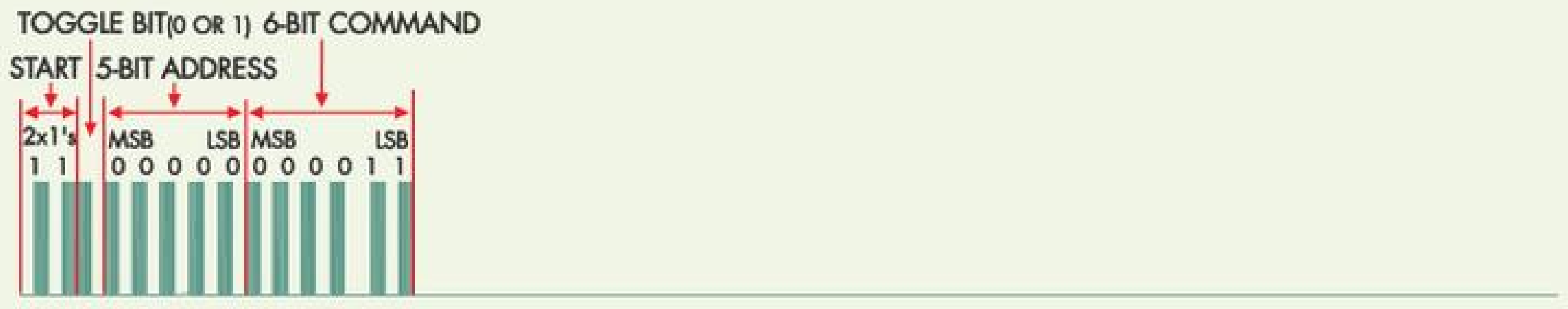
There is also the Pulse Width Protocol used by Sony and Pulse Distance Protocol, originating from NEC.

For more details, see the Freescale Semiconductors application note AN3053: www.nxp.com/docs/en/application-note/AN3053.pdf

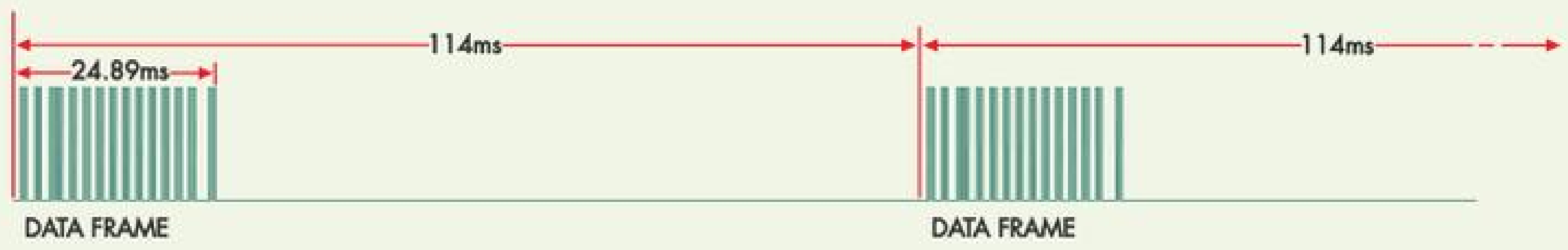
Philips RC5 (Manchester-encoded – 36kHz)


LOGIC 0
LOGIC 1
889µs
889µs
889µs
889µs

LOGIC BIT ENCODING


TOGGLE BIT(0 OR 1)
6-BIT COMMAND
START
5-BIT ADDRESS
2x1's
MSB
LSB
MSB
LSB
1 1
0 0 0 0 0
0 0 0 0 1 1

DATA FRAME SEQUENCE


114ms
24.89ms
114ms
DATA FRAME
DATA FRAME

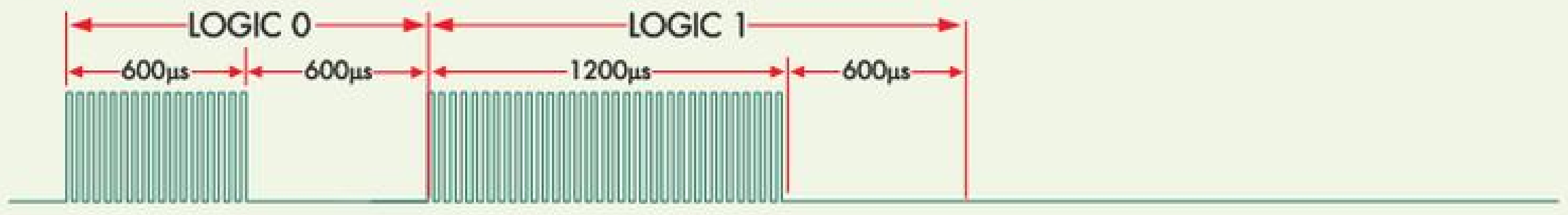
OVERALL SEQUENCE

For this protocol, the 0s and 1s are transmitted using 889µs bursts and pauses at 36kHz. A '1' is an 889µs pause then an 889µs burst, while a '0' is an 889µs burst followed by an 889µs pause. The entire data frame has start bits comprising two 1s followed by a toggle bit that could be a 1 or 0. More about the toggle bit later.

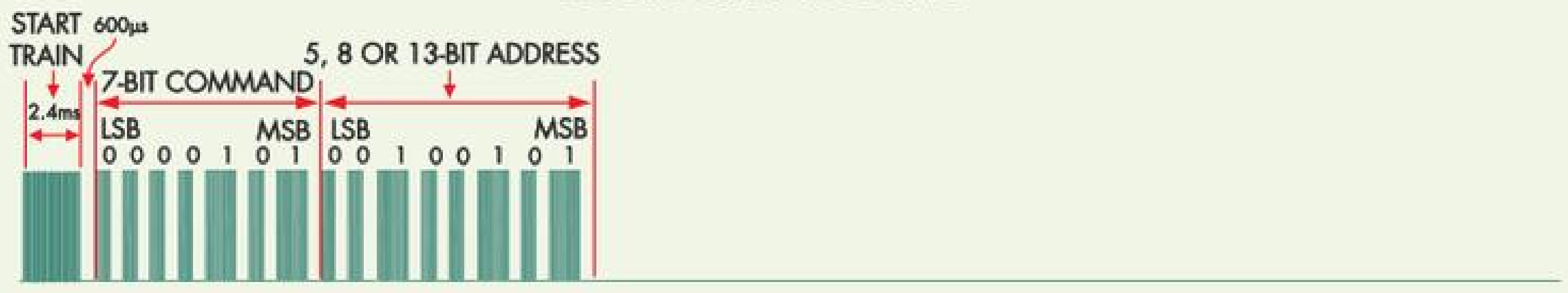
The data comprises a 5-bit address followed by a 6-bit command. The most-significant command bits come first.

When a button is held down, the entire sequence is repeated at 114ms intervals. Each repeat frame is identical to the first. However, if transmission stops, then the same button is pressed again, the toggle bit changes. This informs the receiver as to how long the button has been held down. That's so it can, for example, know when to increase volume at a faster rate after the button has been held down for some time.

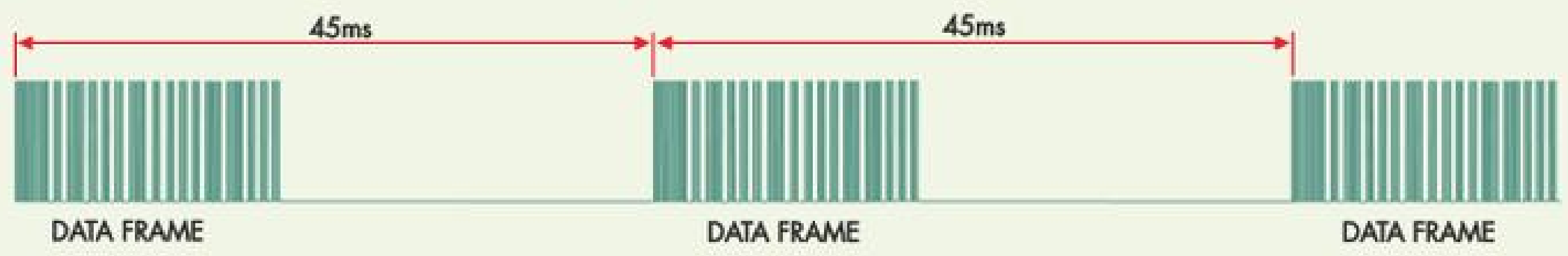
Sony Pulse Width Protocol (40kHz)


LOGIC 0
LOGIC 1
600µs
600µs
1200µs
600µs

LOGIC BIT ENCODING


START TRAIN
600µs
5, 8 OR 13-BIT ADDRESS
7-BIT COMMAND
2.4ms
LSB
MSB
LSB
MSB
0 0 0 0 1 0 1
0 0 1 0 0 1 0 1

DATA FRAME SEQUENCE

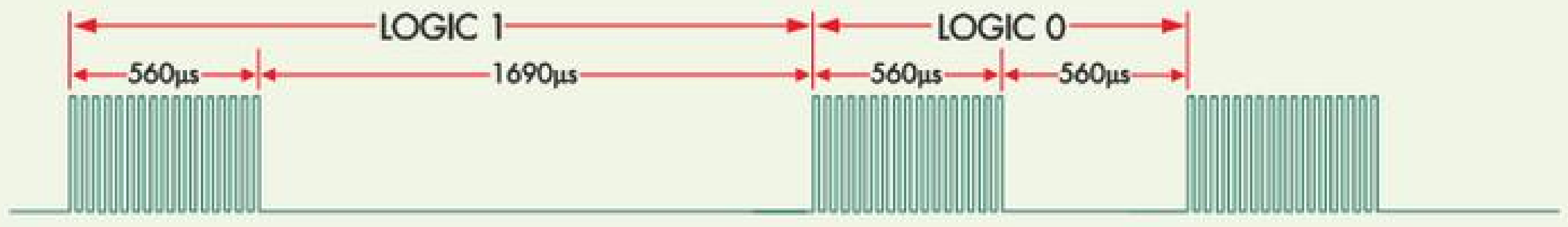

45ms
45ms
DATA FRAME
DATA FRAME
DATA FRAME

OVERALL SEQUENCE

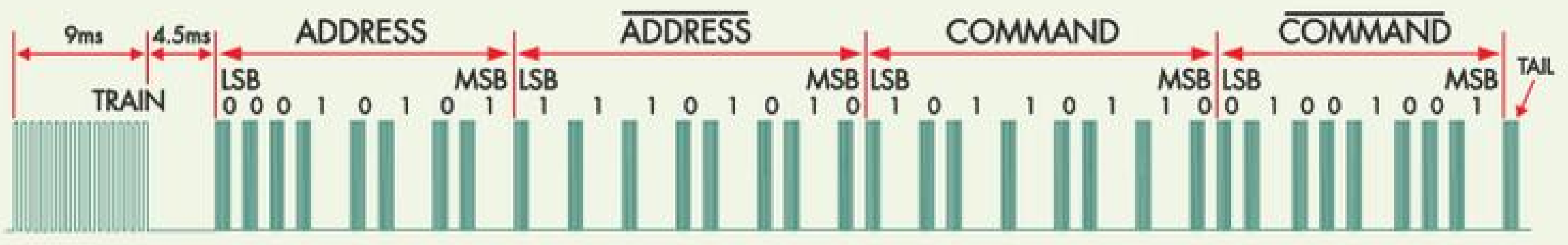
This is also known also as SIRC, which is presumably an acronym for Sony InfraRed Code. For this protocol, the 0s and 1s are transmitted with a differing overall length. The pause period is the same at 600µs, but a '1' is sent as a 1200µs burst at 40kHz, followed by a 600µs pause, while a '0' is sent as a 600µs burst at 40kHz followed by a 600µs pause.

The entire data frame starts with a 2.4ms burst followed by a 600µs pause. The 7-bit command is then sent with the least-significant bits first. The address bits follow (again, least-significant bits first). The address can be 5-bits, 8-bits or 13-bits long to make up a total of 12, 15 or 20 bits of data. Repeat frames are the entire above sequence sent at 45ms intervals.

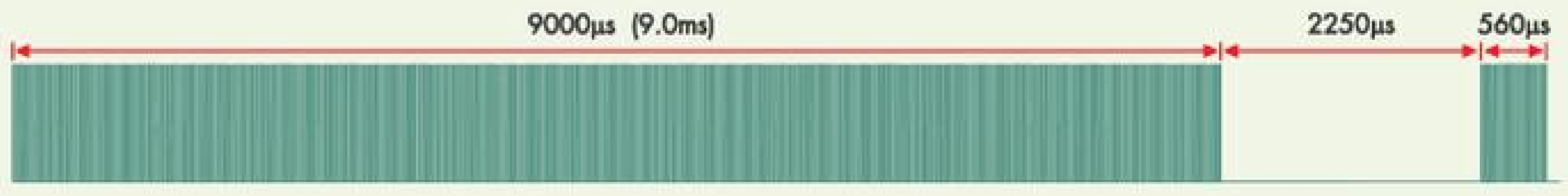
NEC Pulse Distance Protocol (PDP – 38kHz)


LOGIC 1
LOGIC 0
560µs
1690µs
560µs
560µs

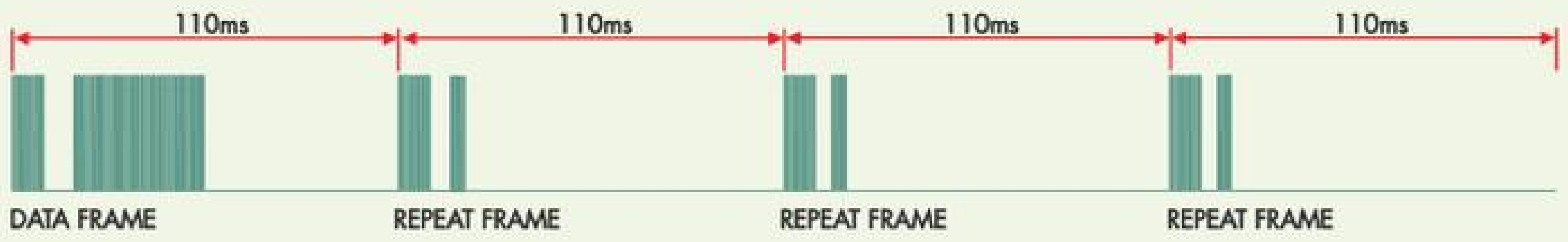
LOGIC BIT ENCODING


9ms
4.5ms
TRAIN
ADDRESS
$\overline{\text{ADDRESS}}$
COMMAND
$\overline{\text{COMMAND}}$
LSB
MSB
LSB
MSB
LSB
MSB
LSB
MSB
0 0 0 1 0 1 0 1
1 1 1 0 1 0 1 0
1 0 1 1 0 1 1 0
0 1 0 0 1 0 0 1
TAIL

DATA FRAME SEQUENCE


9000µs (9.0ms)
2250µs
560µs

REPEAT FRAME


110ms
110ms
110ms
110ms
DATA FRAME
REPEAT FRAME
REPEAT FRAME
REPEAT FRAME

OVERALL SEQUENCE

For the NEC infrared remote control protocol, binary bits zero and one both start with a 560µs burst modulated at 38kHz. A logic 1 is followed by a 1690µs pause, while a logic 0 has a shorter 560µs pause. The entire signal starts with a 9ms burst and a 4.5ms pause.

The data comprises the address bits and command bits. The address identifies the equipment type that the code works with, while the command identifies the button on the remote control which was pressed.

The second panel shows the structure of a single transmission. It starts with a 9ms burst and a 4.5ms pause. This is then followed by eight address bits and another eight bits which are the 'one's complement' of those same eight address bits (ie the 0s become 1s and the 1s become 0s). An alternative version of this protocol uses the second series of eight bits for extra address bits.

The address signal is followed by eight command bits, plus their 1's complement, indicating which function (eg, volume, source...) should be activated. Then finally comes a 560µs 'tail' burst to end the transmission. Note that the address and command data is sent with the least-significant bit first.

The complementary command bytes are for detecting errors. If the complement data value received is not the complement of the data received then one or the other has been incorrectly detected or decoded. A lack of complementary data could also suggest that the transmitter is not using the PDP protocol.

After a button is pressed, if it continues to be held down, it will produce repeat frames. These consist of a 9ms burst, a 2.25ms pause and a 560µs burst. These are repeated at 110ms intervals.

The repeat frame informs the receiver that it may repeat that particular function, depending on what it is. For example, volume up and volume down actions are repeated while the repeat frame signal is received but power off or mute would be processed once and not repeated with the repeat frame.

Trimpot VR1 is next. It has a value of 10kΩ and may be marked as either '10k' or '103'.

Follow that with the LEDs (LED1 and LED2). The anode (longer lead) goes into the hole marked 'A' on the PCB. The LEDs should be installed with the base of their lenses about 5mm above the PCB. Switch S1 can also be fitted now.

Next, solder transistor Q1 to the PCB, with its flat side facing as shown. You may need to bend its leads out (using, for example, small pliers) to fit the pad pattern on the board.

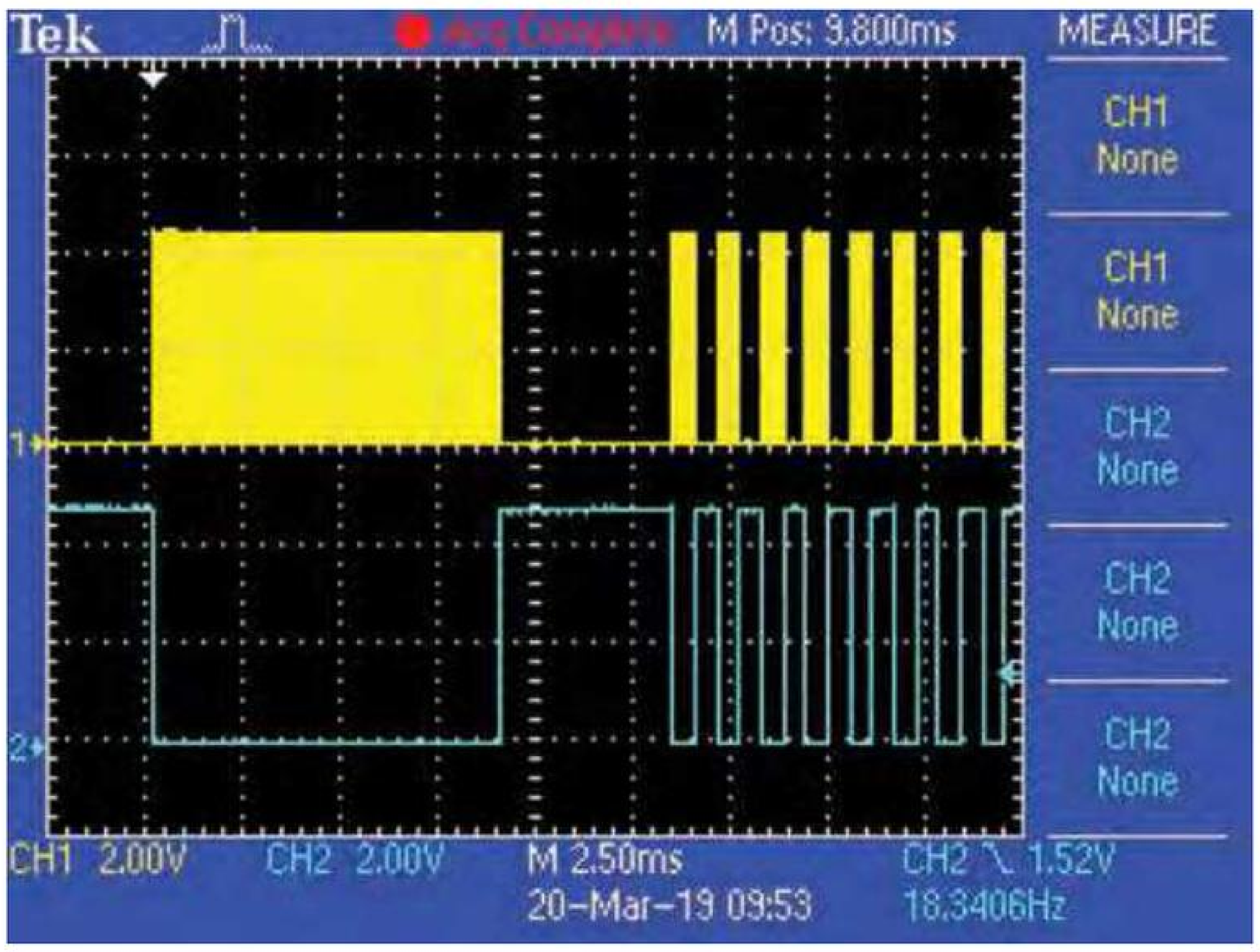


Fig.2 shows the output signals at pin 5 of IC1 (yellow) and the collector of Q1 (cyan), demonstrating the 36-40kHz modulation applied to pin 5 but not on Q1's collector. Note that the collector has a 10kΩ pullup resistor to 5V to be able to show the voltage swing from Q1. Here, the NEC protocol is being used so the modulation is 38kHz.

Now install the two screw terminal blocks. CON1 is mounted with the wire entry holes towards the left-hand edge of the PCB while CON2 should be fitted with the wire entries toward the right-hand edge. You can make up a 4-way terminal by dovetailing two 2-way terminals.

If you are using a socket for IC1 as suggested, plug in the chip now, ensuring that its pin 1 dot is oriented as shown in Fig.3.

Housing it

The *SWC Adaptor* may fit inside the head unit if there is room, or you can mount it outside the head unit in a UB5 box. We used a flanged box that has an extended length lid with extra mounting holes. This makes it easier to mount in the car, under the dashboard is the logical location.

Alternatively, a standard UB5 box can be used instead, or the unit can be wrapped in insulation and cable tied in position.

If fitting it into a box, drill holes at either end to fit the cable glands which allow the power supply and infrared LED wiring to pass through.

There are cut-outs in the PCB to accommodate the gland nuts which go inside the box. But note that these nuts must be oriented correctly, with two of the sides vertical, so they will fit into the recesses in the board.

The PCB is mounted in the box on four 12mm-long M3 tapped spacers, using eight machine screws. Mark out and drill the 3mm holes for PCB mounting while you are making the holes for the cable glands.

Installation

The *SWC Adaptor* is wired into the vehicle so that it gets +12V power when the ignition is switched on. Virtually all head units have connecting wires carrying 0V (GND) and ignition-switched +12V, so you can tap into the supply there.

Just make sure the +12V wire has power with the ignition on and not with the ignition off.

The SWC input on the *SWC Adaptor* connects to the steering wheel control wire. You should already know where to tap into it from the previous test where you determined that your steering wheel controls are suitable for use with this unit.

The *SWC Adaptor* has two pairs of output wires: one pair to drive an external infrared LED (LED3) and another connecting to the collector and emitter of the transistor which provides the unmodulated output. You can use either to control the head unit. Each option has advantages and disadvantages.

The infrared LED approach has the advantage that the head unit does not need to be opened up; the infrared LED is simply placed over the infrared receiver on the head unit. The disadvantage is that the wiring to this LED, and the LED itself, will be visible.

The easiest way to do this is to use a premade IR Remote Control Extension Cable. These are available from Jaycar (see parts list). This has an infrared LED already mounted in a small neat housing, with a long lead.

You will need to figure out how to route that cable from the *SWC Adaptor* mounting location to the IR receiver on the head unit.

Adhesive wire saddles are useful for keeping this wiring neat.

The Jaycar IR extender has a 3.5mm jack plug which you can cut off, as it isn't needed. The LED anode wire is the one which was connected to the jack plug tip. You can also get similar extenders from eBay or AliExpress, most of which have bare wire ends. Whichever one you use, wire it to the A and K terminals of CON2.

It's then just a matter of sticking the LED emitter package to the front of your head unit, directly in front of the infrared receiver, using its own self-adhesive pad.

If you do not know where the infrared receiver is, it will be in an area free from switches and knobs.

The front panel may have a purple-looking area over the infrared receiver, different in appearance from the rest of the panel.

If you still can't figure it out, you will need to test the unit while moving the transmitter around the panel until you find a location where it works reliably. You can then stick it in place.

Tweaking the button sensing

Once you have the unit wired up to power and the steering wheel controls, it is a good idea to perform some checks to make sure it is sensing the steering wheel buttons accurately.

The *SWC Adaptor* button-sensing input includes a 1kΩ pull-up resistor to 5V. This is shown with an asterisk both on the circuit and PCB. This resistor may need to be changed in some vehicles to give reliable button detection and discrimination.

To check it, monitor the voltage between TP GND and TP2 when the unit is powered up, pressing each steering wheel button in turn.

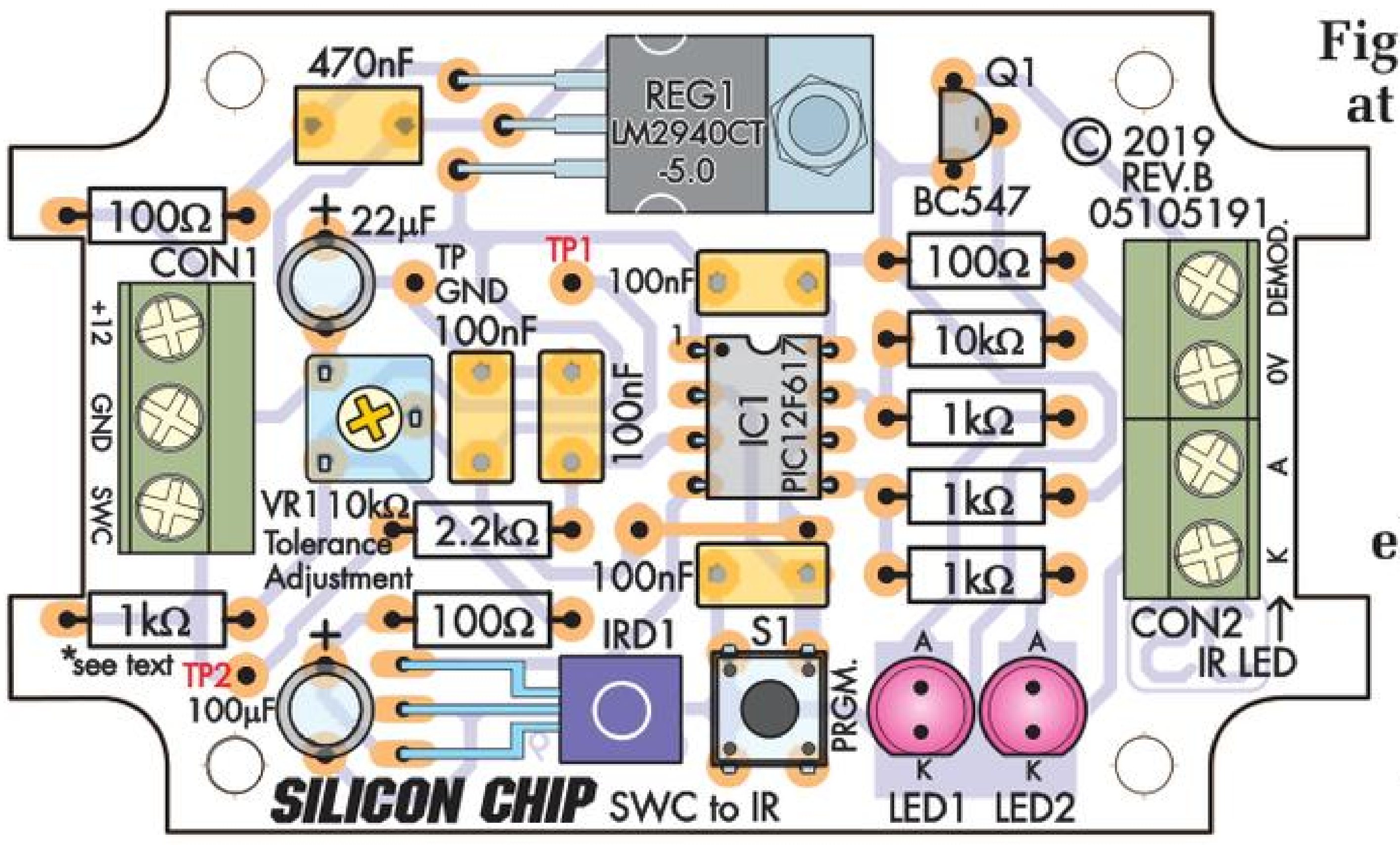
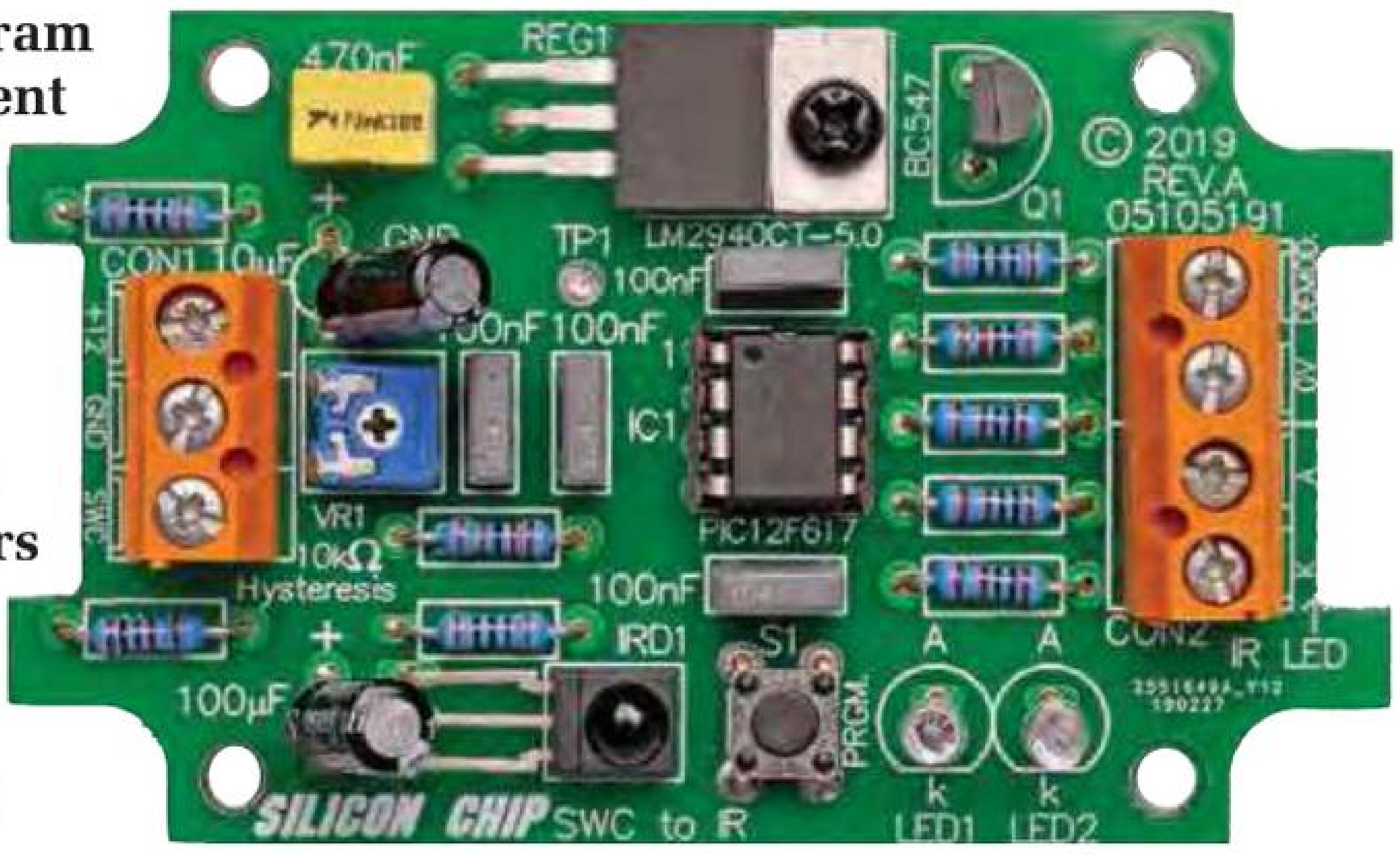


Fig.3: the overlay diagram at left shows component placement while the matching fully component installed PCB is shown at right. Make sure the two electrolytic capacitors and IC1 are correctly oriented with the shown polarity.



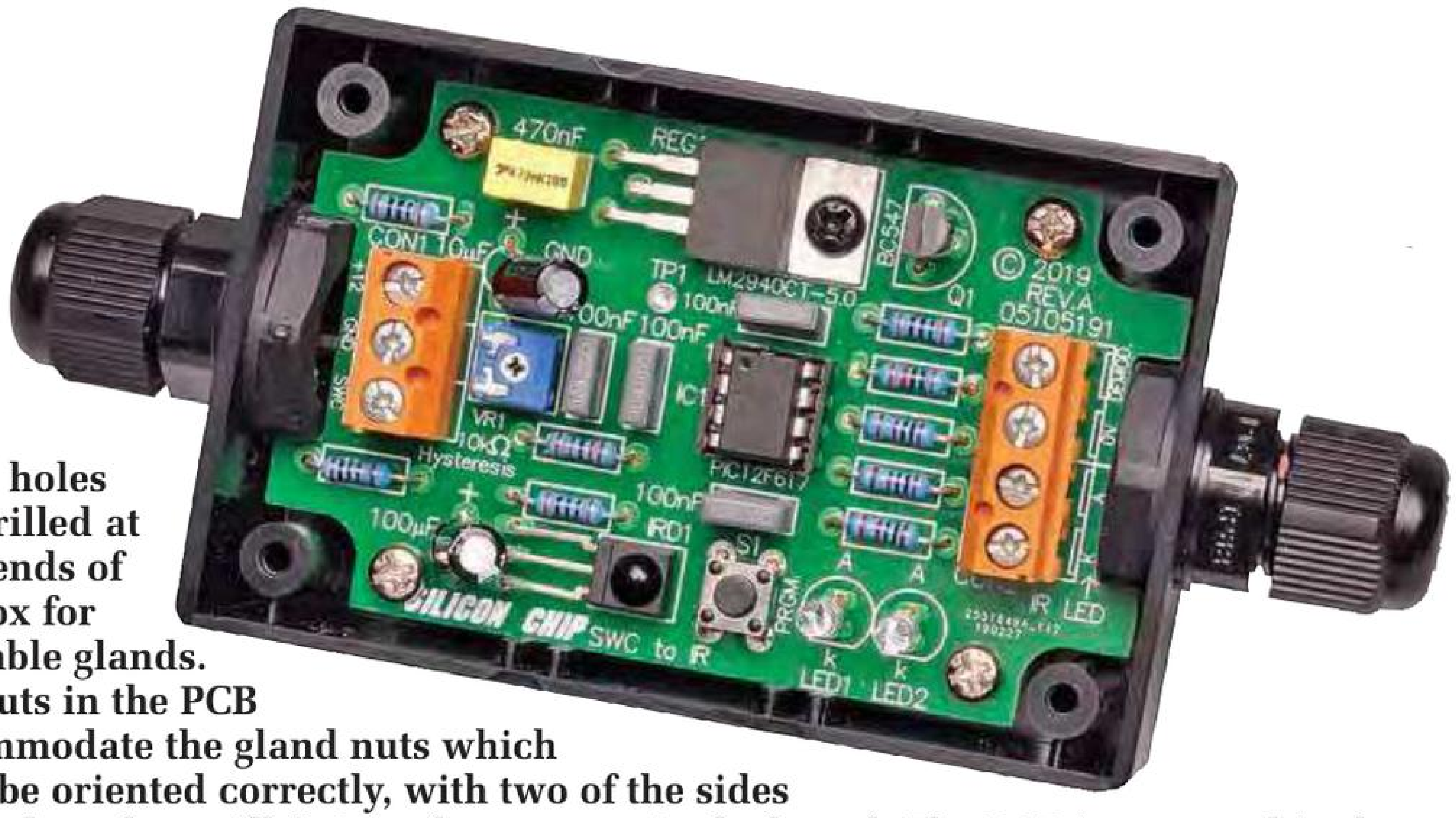


Fig.4: holes are drilled at both ends of the box for the cable glands. Cut-outs in the PCB accommodate the gland nuts which must be oriented correctly, with two of the sides vertical, so they will fit into the recesses in the board. The PCB is mounted in the box on four 12mm-long M3 tapped spacers and attached using M3 screws

On our test vehicle, we measured 3.93V with switches open, then 0.383V, 0.708V, 1.11V, 1.59V, 2.2V and 2.98V when each of six switches was pressed individually. So we had reasonable steps of more than 300mV between each voltage. The unit's tolerance should then be set to half that value; in this case, 150mV or less. So we adjusted VR1 for 1.5V at TP1.

But we could have improved the voltage range if the 1kΩ resistor was changed to 510Ω. That would give 4.37V with switches open and 0.67V, 1.19V, 1.77V, 2.34V, 3.02V and 3.7V with each pressed individually. That would give us a minimum step of at least 500mV and so the tolerance value could be set to 250mV (2.5V at TP1). But as long as the tolerance can be set to at least 100mV (ie, at least 200mV between the two closest voltage readings), we would consider that acceptable.

– Parts List – Steering Wheel Control Adaptor

- 1 PCB coded 05105191, measuring 77 × 47mm, from the *PE PCB Service*
- 1 UB5 Jiffy box (optionally with flange)
- 1 3-way PCB mount screw terminal with 5.08mm spacing (CON1)
- 2 2-way PCB mount screw terminals with 5.08mm spacing (CON2)
- 1 DIL-8 IC socket
- 1 momentary SPST pushbutton switch [Altronics S1120, Jaycar SP-0600] (S1)
- 9 M3 × 6mm pan head machine screws
- 1 M3 hex nut
- 4 M3 tapped × 12mm spacers
- 2 IP65 cable glands for 3-6.5mm wire

Semiconductors

- 1 PIC12F617-I/P microcontroller programmed with 1510519A (IC1)
- 1 LM2940CT-5.0 5V automotive regulator (REG1)
- 1 Infrared receiver [Jaycar ZD1952 or ZD1953, Altronics Z1611A] (IRD1)
- 1 BC547 NPN transistor (Q1)
- 2 3mm high brightness red LEDs (LED1,LED2)
- 1 Infrared Remote Control Receiver Adaptor Extender Extension Cable [Jaycar AR1811 or similar] with adhesive backing for direct mount over IR sensor (LED3)

Capacitors

- 1 100µF 16V PC electrolytic
- 1 22µF 16V PC electrolytic
- 1 470nF 63V MKT polyester (*code 474, 0.47 or 470n)*
- 4 100nF 63V MKT polyester (*code 104, 0.1 or 100n)*

Resistors (0.25W, 1%)

- 1 10kΩ
- 4 1kΩ
- 1 2.2kΩ
- 3 100Ω
- 1 10kΩ miniature horizontal mount trim pot (VR1) *(may have code 103)*

Miscellaneous

Automotive wire, solder, connectors, self tapping screws etc.

If your steering wheel control switches provide a voltage range that differs significantly from ours, you may benefit from adjusting the 1kΩ resistor value. If your voltage readings are mostly low, try using a lower value, while if your readings are all on the high side, try using a higher value. But don't go below 200Ω as you then risk damaging the resistors in your steering wheel.

Using the unmodulated output

The advantage of using the unmodulated output from the *SWC Adaptor* is that it can be wired internally to the head unit, so the wiring may be able to be hidden. Usually, only a single wire needs to be connected to the infrared receiver on the head unit. This wire can pass out the back of the head unit and routed to the *SWC Adaptor*.

The disadvantage of this approach is that you need to open up the head unit, find the infrared sensor output and solder the wire to it. How this is done is best shown in the accompanying photos opposite.

In Fig.6, we've opened up the front panel of the head unit and located the infrared receiver (arrowed). But this is not the best location to connect the wire.

Fig.5 shows the multiway connector which is used to connect the front panel to the head unit.

To figure out which pin carried the infrared receiver signal, we plugged the front panel back into the head unit and opened its case, then located where the front panel connector is terminated (see Fig.7). We then powered it up using a 12V DC source and connected a DMM set to measure volts between 0V and each pin at the rear of the front panel in turn.

Look for a pin which measures around 5V, then measure its voltage while an infrared transmitter is placed in front of the unit and a button held down, so it is transmitting. If you have the correct pin, that voltage reading should drop slightly while the infrared remote control transmitter is active. In our case, we found that it dropped from 5V to 4.75V during infrared reception.

The arrowed pin in Fig.7 is the one that we determined carries the infrared signal, and this is where we soldered the wire.

You could use an oscilloscope to look for the pulses from the infrared receiver; however, the multimeter method is easier and generally works well.

The *SWC Adaptor* output includes a 0V connection for the unmodulated

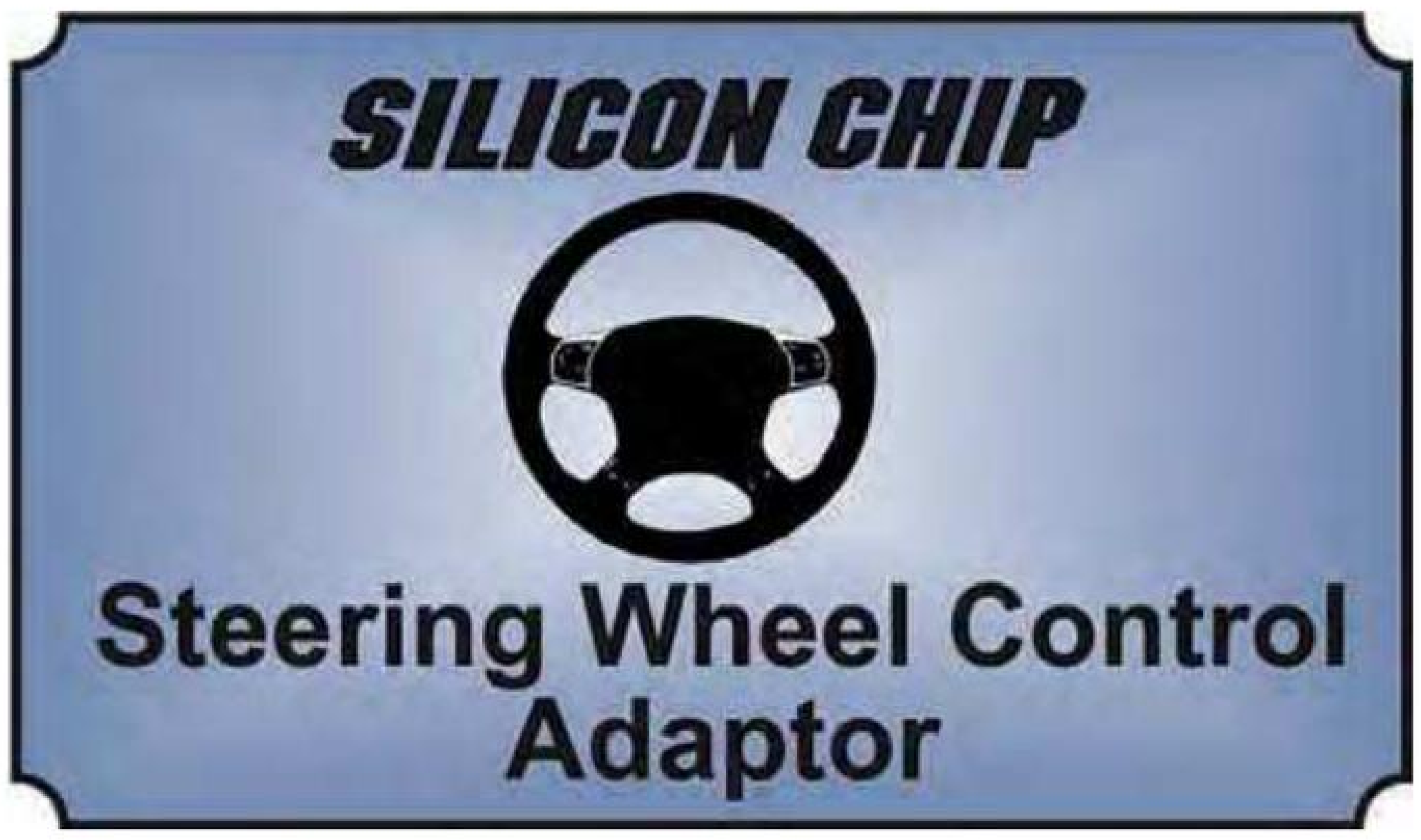


Fig.8: (not to scale) the front panel for the *SWC Adaptor* can be downloaded as a PDF from the August 2020 page of the *PE* website and printed onto paper, transparent film or adhesive-backed vinyl.

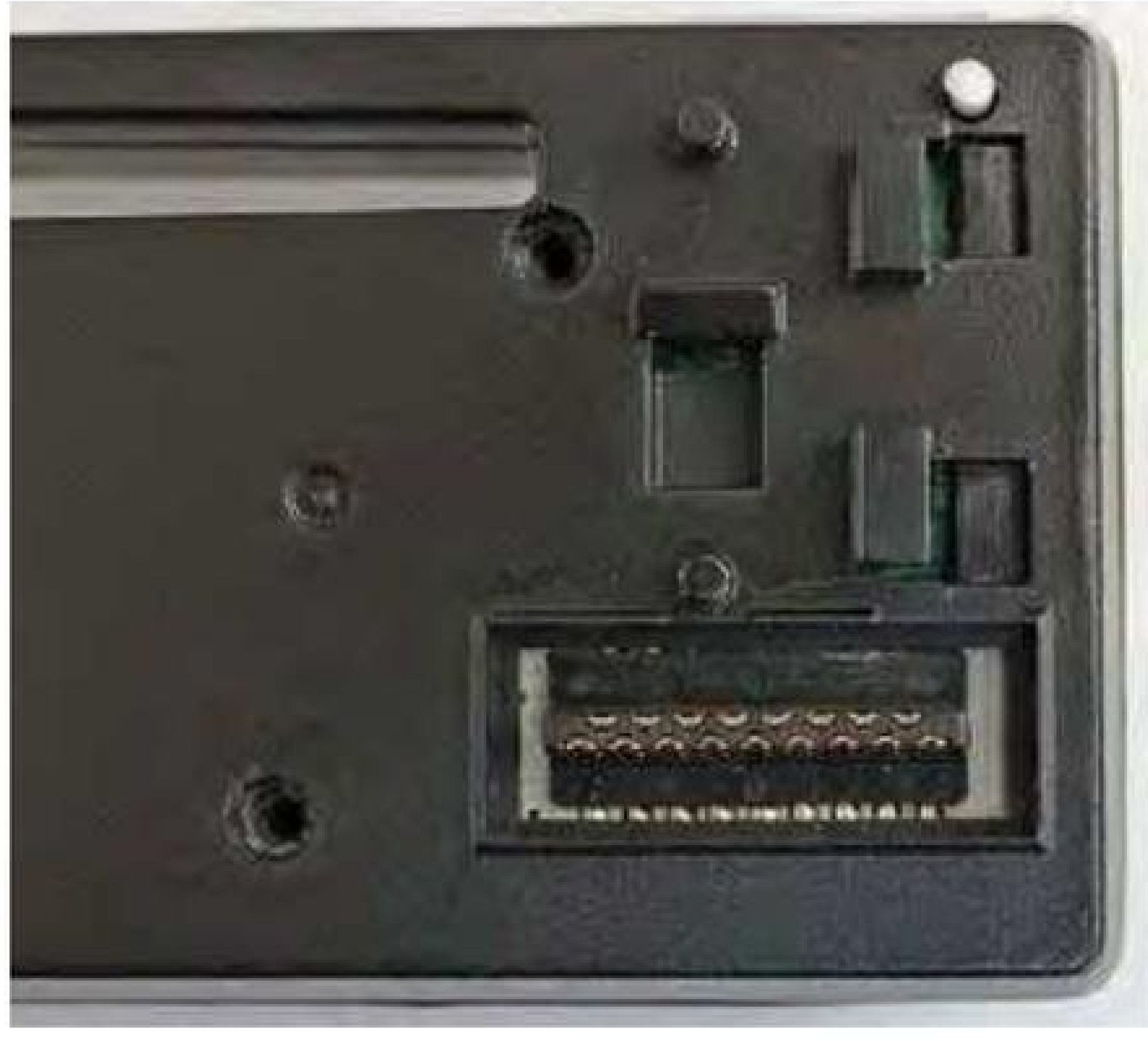
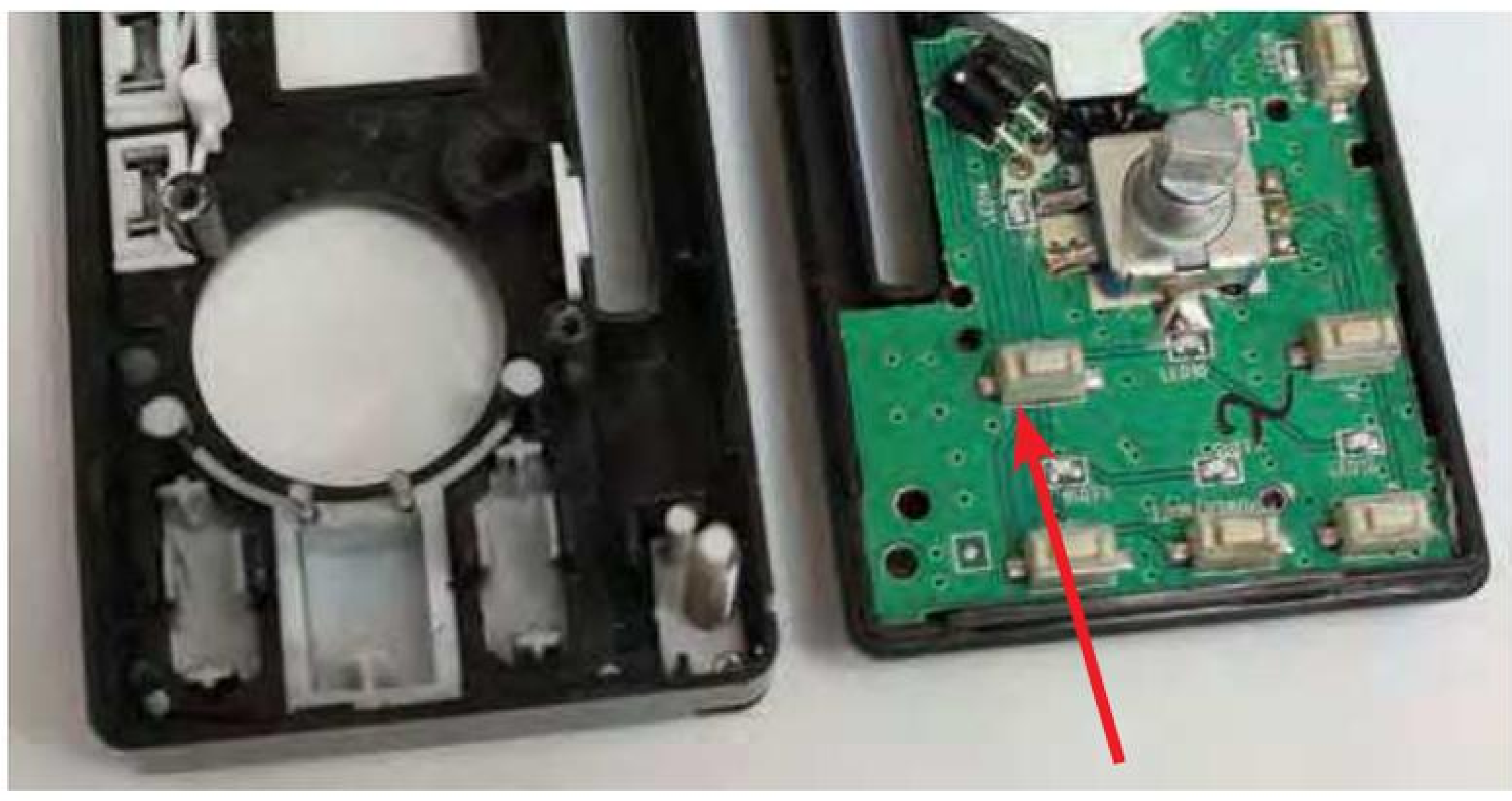


Fig.5 (left) the multiway connector which is used to connect the front panel to the head unit.

Fig.6 (right) shows the head unit's opened-up front panel and the location of the infrared receiver (arrowed). *But*, this is not the best location to connect the wire (see below).



output. This can be wired to a ground connection on the same multi-pin connector. However, this should not be necessary as the infrared receiver on the head unit should have its ground pin connected to the head unit chassis and would be at the same potential as the 0V connection on CON1.

If you have problems with the unmodulated connection working, try connecting a wire between these two points to see if that solves it.

Setting up the unit

Now you need to decide what functions you want from each switch on the steering wheel. Typically, this would include volume up and down, source selection, next and previous file/track/frequency/station and power on/off.

You are not restricted to the original purposes of each switch, although it would be less confusing to do so. You can use each switch to perform any of the functions available on the handheld remote control supplied with your head unit.

For some buttons, you may want the function to repeat if held down (eg, volume up/down) but with others, you may not (eg, source selection or on/off).

We found with some head units, holding down the source selection button would result in nothing happening. You would have to press the button only for a short period to switch to the next source. That's not ideal when using steering wheel buttons. So we have included a feature in the *Adaptor* where two out of the 10 possible buttons will not generate repeat codes even if held down.

So it's just a matter of assigning functions which may have this shortcoming on your head unit to those two button positions. This would generally include source selection, power on/off, radio band change or mute. None of these need the repeat function.

You can test whether this is necessary by holding those buttons down on your infrared remote control and seeing whether the unit behaves as desired, or not.

Programming the button functions

You can now match up the voltages produced by each steering wheel button to the desired infrared function. You can program up to 10 switches. It does not matter what order you program each switch, and you don't have to use all 10. The non-repeat feature mentioned above applies to switches nine and 10, so you can skip some positions if you don't have 10 buttons but need this feature.

All of the programmed infrared codes must use the same infrared protocol (NEC, Sony and RC5 are supported – see the *Infrared Coding* panel).

That should not be a problem given that your head unit remote control will be using one protocol for all of its buttons – and most likely, one of those supported by this unit.

To enter the programming mode, hold down S1 while switching on the vehicle ignition. Entering programming mode clears any previous programming.

So you must program the functions of all switches each time this mode is invoked. Upon the release of S1, LED1 will flash once, indicating that the *SWC Adaptor* is ready to program the first switch function.

Point the handheld remote toward the infrared receiver on the *SWC Adaptor* and press the required function button. LED2 should light up. If it does not, it is possible that your handheld remote does not use one of the three supported protocols. LED2 will light up continuously for codes received in the NEC protocol. It will flash off once and then on for the Sony protocol and flash off twice for RC5.

Now press and hold the steering wheel switch that you want to assign to that function, then press S1 on the *SWC Adaptor*. The input voltage for that switch and the infrared code will then be stored in permanent Flash memory for that switch position. LED1 will then flash twice, to indicate that the *Adaptor* is ready to accept the infrared code for the second switch function.

Continue programming each switch for the function required. Each time you press S1, LED2 will flash a certain number of times, indicating the next switch number that is ready to be programmed. You can press S1 again to skip a position that you don't want to assign (eg, if you have less than ten steering wheel buttons). Once the tenth position is programmed, the *SWC Adaptor* will stop and not respond.

Switch off power and when you then switch it back on again, without pressing S1 on the unit, the *SWC Adaptor* will begin normal operation, reproducing the stored infrared code each time one of the selected steering wheel buttons is pressed. This also applies if you don't program all ten positions; merely switch off the ignition when you have finished programming all the functions that are required.

To use the special non-repeat feature at positions nine and ten, you can skip over the earlier positions using extra presses of S1 to reach them if you are not programming all 10 functions.

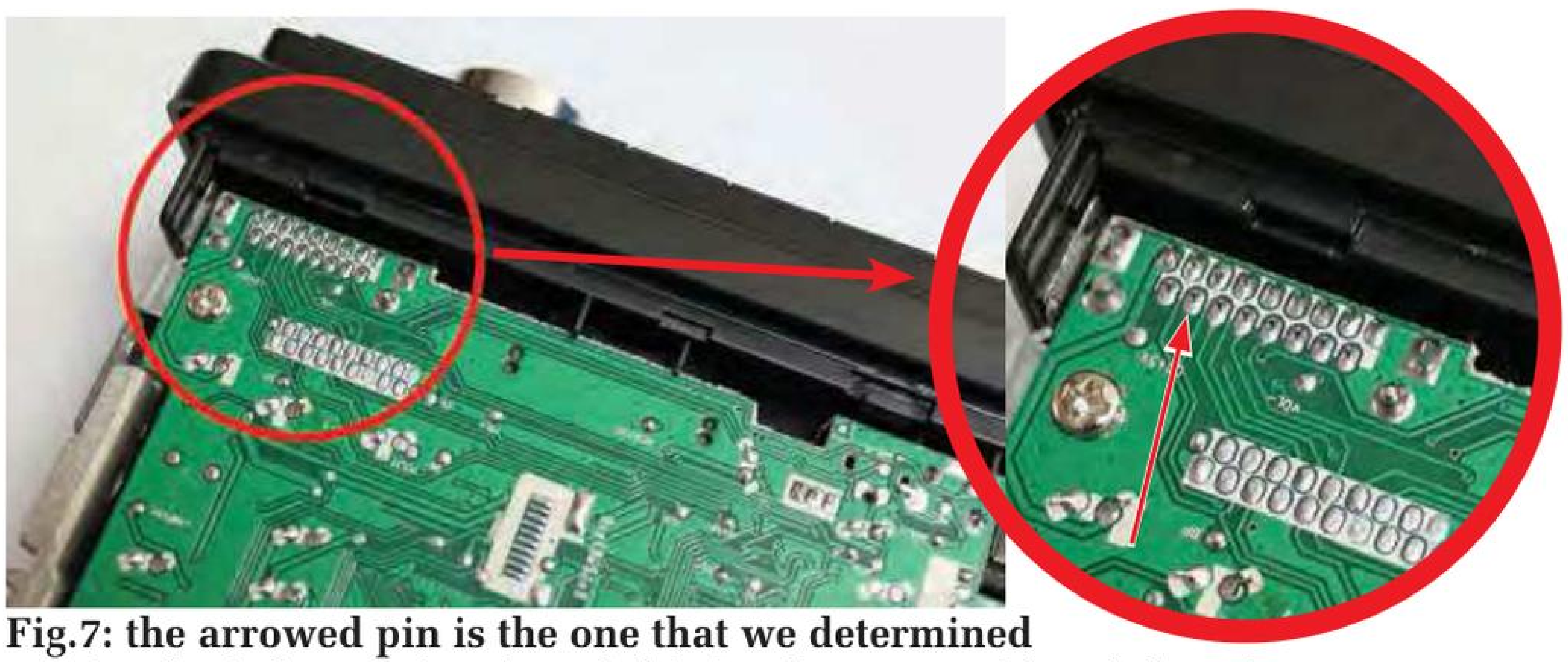


Fig.7: the arrowed pin is the one that we determined carries the infrared signal, and this is where we soldered the wire.

Reproduced by arrangement with
SILICON CHIP magazine 2020.
www.siliconchip.com.au

NO JUNK MAIL
ADDRESSED MAIL ONLY

SIGNS DON'T WORK! YOU NEED THIS JUNK MAIL REPELLER!

by Allan Linton-Smith

Is your letterbox full of junk, even though you have a NO JUNK MAIL sign? If so, you need to build our *Junk Mail Repeller*. It might not completely prevent junk mail from being shoved in your box. . . but it should at least help. And you'll have some fun watching the reactions of the would-be junkmeister!

Let's face it, the people who deliver junk mail must be completely blind (or unable to read!) because they can't seem to understand the 'NO JUNK MAIL' or 'NO ADVERTISING MATERIAL' sign in giant letters on your letterbox.

But hopefully they aren't deaf, too; that's where this gadget comes in. For a little over a pound (plus a few bits and pieces that you probably already have), you can build this junk mail-triggered digital audio recorder/playback device.

Just imagine, as they cram yet another flyer into your letterbox, a voice yells back at them: 'HEY YOU! The sign says NO JUNK MAIL!'

That's just one of its many uses. But fundamentally, it's just a fun project that you could probably think of a thousand uses for.

Maybe a switch on your bedroom door and a voice saying *'sisters (*or *brothers) not welcome!'*?

By the way, even with a 'NO JUNK MAIL' sign, it isn't illegal for a business or individual to put a flyer in your letterbox (even if it is against the industry code of practice).

The problem lies with psychology 101: the junk they're delivering to you isn't junk – it's a vital message that you would be most upset not to receive.

Therefore any sign doesn't apply to them. Only to the next bloke with *junk*!

So people who stuff junk mail in your box can't be arrested! But you can discourage (and probably annoy) them with this device.

If you actually like and want junk mail (and that is about the only mail you get these days), do not attempt this project.

Or maybe you *should* build it and use it to say 'thank you' to the people delivering free catalogues.

How does it work?

Every time a flyer or catalog goes into your letterbox, the extra weight should be enough to trigger a microswitch – and they're greeted with a message – eg, *'No junk mail please – Royal Mail only......we are watching you!'*

Then have some fun watching their reactions! (Tough luck if it is your postie delivering the junk, as they sometimes do!).

You can put any message you like, in any language. We discourage a tirade of swear words – although that would of course be possible – as it may land on inappropriate ears.

There isn't much to it; it's made from a pre-built, low-cost digital voice recorder which is installed in a plastic box, along with a microswitch and a battery. It then becomes a junk mail repeller!

Description

The voice recorder/playback module we're using is based on an ISD1820 IC, which can record up to 11 seconds of audio.

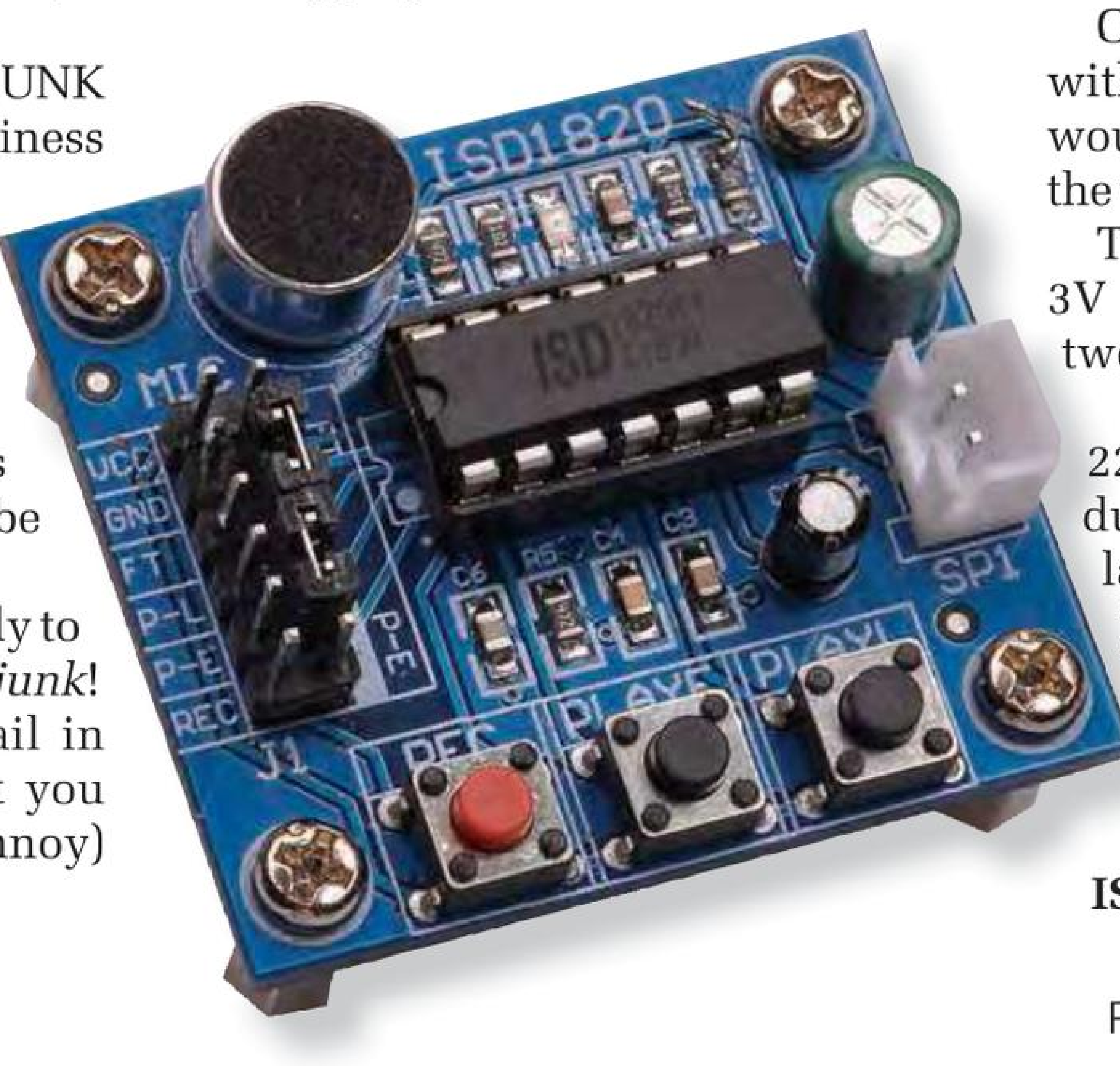
We chose this one because (a) it's a nice, small unit, measuring just 38 × 42.5mm; but (b) more importantly, it's cheap and really easy to get. You can readily source it from eBay or AliExpress – just do a search for 'isd1820'.

We'd suggest being just a little careful on line – the best price we found at the time of writing was £1.74 including postage (AliExpress item 2009871627).

Ours came ready-made, complete with a tiny loudspeaker. The speaker would cost you more than we paid for the whole thing if you bought it locally!

The module can be powered from 3V (its stated maximum is 7V) from two AA cells in series.

The standing current drain is 220µA, and it consumes about 38mA during playback. The cells should last for months, depending on your junk mail load!



The project isn't just based on the ISD1820 module . . . it is the project!

We built our *Junk Mail Repeller* into a UB3 Jiffy box but just about any enclosure will do, as long as it fits inside your letterbox. The microswitch glued to the outside of the lid is the secret: it triggers the voice message whenever anything heavier-than-an-envelope (eg, junk mail!) lands on it. The switch on the end is optional: it changes the length (and quality) of the voice recording which you make to suit the situation.

Note that there is a slightly different module available than the one we used, which has a 10-pin header and two slide switches instead of a 12-pin header.

This one is also suitable for use in this project, but you have to make a few slight changes. These are simple enough that we'll leave them to you. That alternative module is quoted as working from 2.4-5.5V, which is fine since our battery is around 3V.

The circuit

The circuit of our module is shown in Fig.1. The ISD1820 (IC1) is responsible for all audio recording and playback tasks. A 100nF capacitor bypasses its 3V supply (from two AA cells).

During recording, it samples audio from onboard electret microphone MIC1, which is AC-coupled to its pin 4 and 5 differential inputs. MIC1's power supply voltage is filtered by the 1kΩ resistor and 220µF capacitor, while the 4.7kΩ series resistors provide suitable biasing.

A 4.7µF capacitor sets the time constant for IC1's internal automatic gain control (AGC), used during recording to automatically provide a suitable gain for the microphone. Recording is initiated by the REC pin (pin 1) going high and continues as long as it stays high. During recording, the RECLED pin (pin 13) is held low, so LED1 lights.

The RECLED output is also pulsed low at the end of playback, causing LED1 to flash briefly.

IC1 has a small internal audio amplifier, allowing it to drive the 8Ω speaker directly, via pin header CON2. The module is supplied with a suitable cable to connect the speaker to this JST header. Playback is initiated by bringing either pin 2 (PLAYE) or pin 3 (PLAYL) high.

The difference is that the recorded message will continue to play until the end even if PLAYE goes low again, whereas PLAYL must be held high for playback to continue.

In other words, PLAYE is edge-triggered while PLAYL is level-triggered (hence the names). If pin 12 (FT) is held high, audio from the microphone is fed through to the output.

The module has on-board tactile pushbuttons which pull the PLAYE, PLAYL or REC pins high when they are pressed.

These signals are also fed through to pins 7, 9 and 11 of CON1 where they can be connected to, for example, external buttons or microcontroller outputs.

FT is fed to pin 5 of this header, while power and ground appear on pins 1 and 3 respectively.

The other half of CON1 is intended so a jumper can be placed across pins 2 and 4, permanently enabling feedthrough, or between pins 4 and 6, in which case no connection is made and feedthrough is controlled by pin 5.

Bridging pins 10 and 12 causes the RECLED output to be connected to the PLAYE input. Since RECLED is pulsed briefly low at the end of playback, after playback finishes, this will cause playback to start again, as there is a low-high transition on the PLAYE input.

Therefore, playback will loop forever, or at least until the bridging jumper is removed (it can be kept on pins 8 and 10 when not used).

Finally, the 100kΩ resistor from ROSC to ground sets the audio sampling rate to 6.4kHz, which means the maximum length of the audio recording is 10-ish seconds (we measured it at 11). This can be changed either to give a longer recording time with worse quality, or a shorter time with better quality.

Chip internals

Fig.2 shows the internal block diagram for the ISD1820 IC. It comprises a microphone preamplifier, oscillator, audio sample storage array, audio amplifier, filters, power conditioning and control logic.

The storage array is quoted as retaining the saved audio data for up to 100 years, or until the next time you press the REC button.

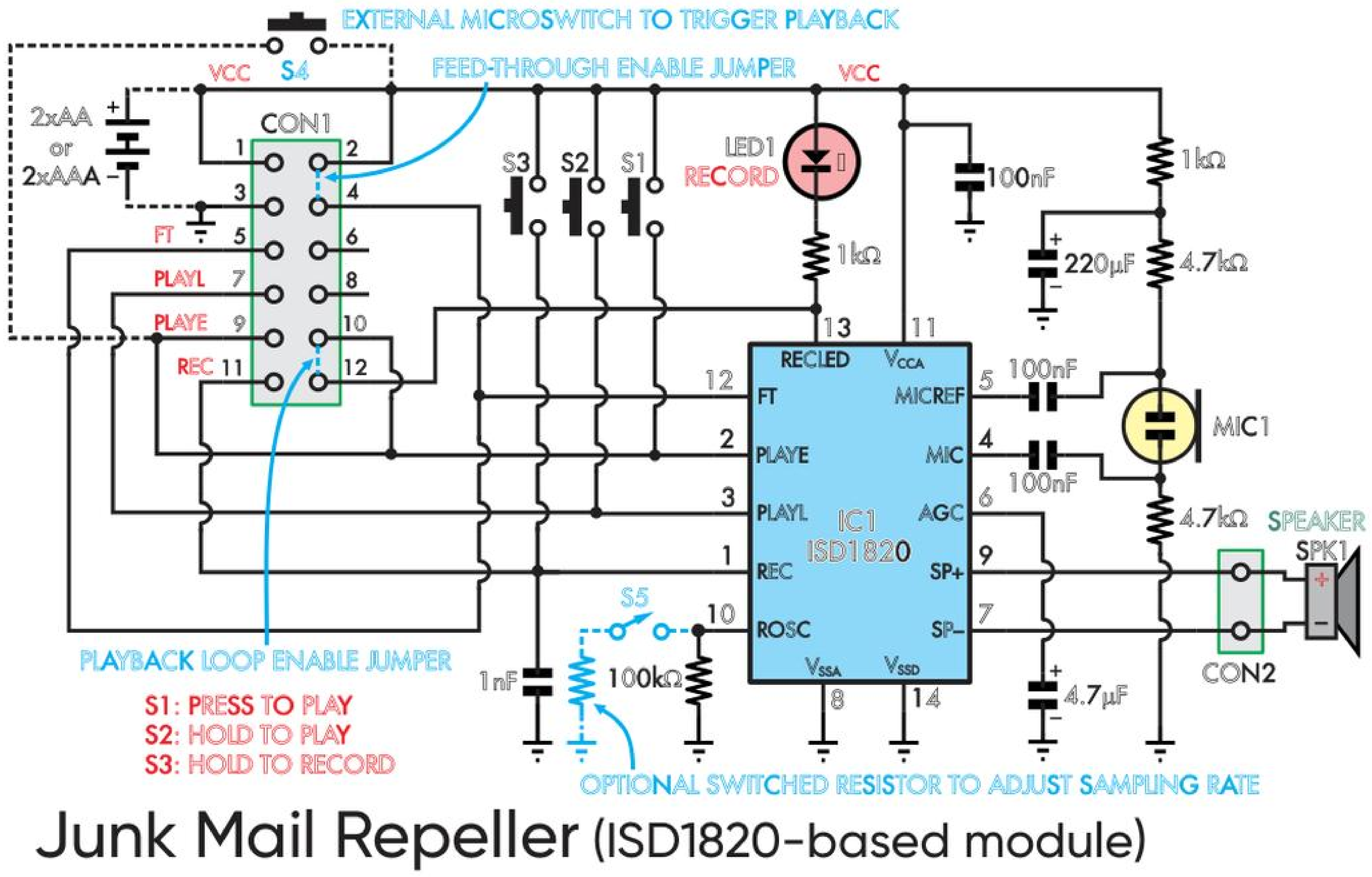

EXTERNAL MICROSWITCH TO TRIGGER PLAYBACK
VCC
S4
FEED-THROUGH ENABLE JUMPER
VCC
2xAA
or
2xAAA
CON1
1
2
3
4
FT
5
6
PLAYL
7
8
PLAYE
9
10
REC
11
12
S3
S2
S1
LED1
RECORD
100nF
1kΩ
1kΩ
220µF
4.7kΩ
13
11
RECLED
VCCA
12
FT
5
MICREF
100nF
MIC1
2
PLAYE
4
MIC
100nF
3
PLAYL
IC1
ISD1820
6
AGC
4.7kΩ
SPEAKER
SPK1
1
REC
9
SP+
S5
10
ROSC
7
SP–
PLAYBACK LOOP ENABLE JUMPER
1nF
100kΩ
VSSA
VSSD
8
14
4.7µF
CON2
S1: PRESS TO PLAY
S2: HOLD TO PLAY
S3: HOLD TO RECORD
OPTIONAL SWITCHED RESISTOR TO ADJUST SAMPLING RATE
Junk Mail Repeller (ISD1820-based module)

Fig.1: the circuit of the voice recorder/playback module, with IC1 providing all of the recording and playback functions. This diagram includes the three extra components you will need, ie, a two-cell battery to power the unit, a microswitch to trigger playback of the recorded audio and optionally, a resistor connected across the onboard 100kΩ resistor to provide better sound quality.

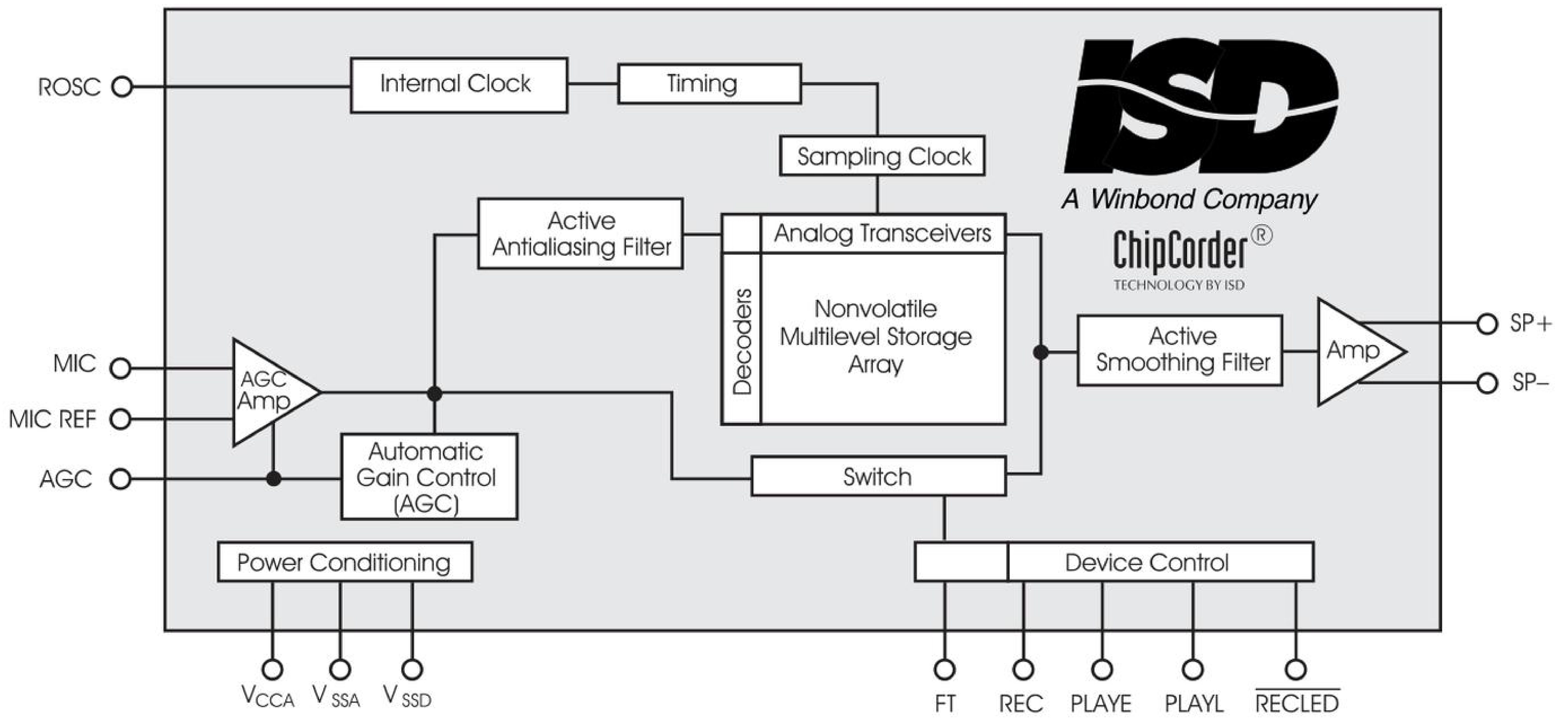


Fig.2: the internal workings of the audio recording and playback chip. The external resistor from ROSC to ground sets the oscillator frequency which determines the sampling rate. When recording is activated, the output of the microphone preamp feeds into the storage array via an antialiasing filter. And when playback is activated, the contents of the storage array are fed to the output amplifier, which is capable of driving an 8Ω speaker at a reasonable volume.

The power amplifier can deliver about 80mW into 8Ω, which is sufficient to give quite a reasonable volume when the speaker is mounted in a Jiffy box (ie, using it as a baffle).

A more powerful amplifier could be hooked up to the output, along with a larger speaker, but this may annoy your neighbours.

Recording quality vs time

We tested various values for the resistor from ROSC to ground and plotted the results in Fig.3. As you can see, it's very close to being a straight line.

The minimum recommended value is 80kΩ, giving a sampling rate of 8kHz and a maximum recording time of eight seconds. But you can reduce the value down to 18kΩ, giving just under three seconds of recording time, and presumably a sampling rate of around 35kHz.

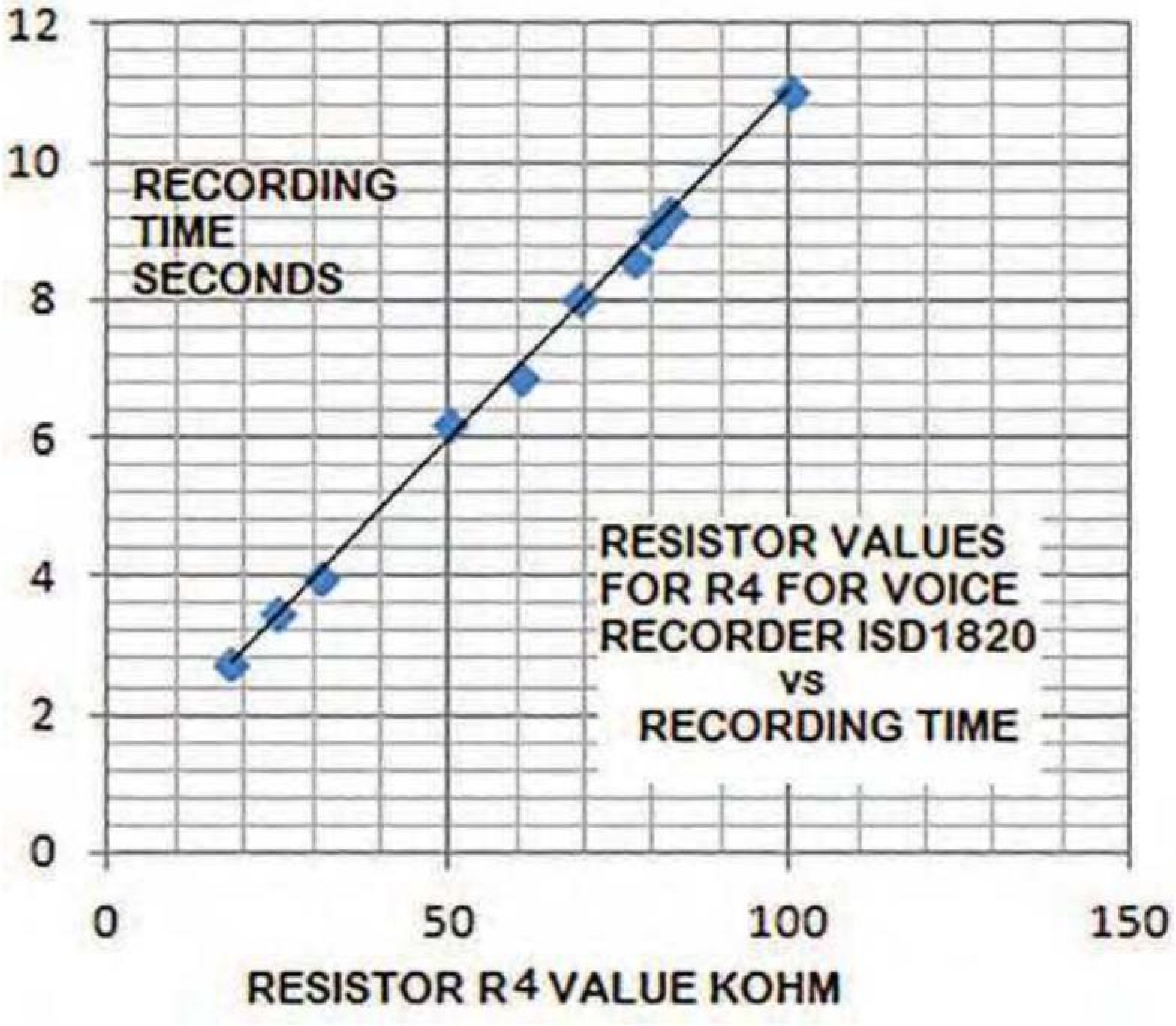


Fig.3: we varied the value of ROSC and measured the recording/playback time. As expected (based on what it says in the data sheet), the sampling rate is inversely proportional to the resistor value, thus the recording time is directly proportional to it. The sampling rate is equal to 640,000 divided by ROSC in kilohms, which gives 6.4kHz with the default value of 100kΩ.

The maximum recommended value is 160kΩ, giving a sampling rate of 4kHz and a maximum playback time of 16 seconds.

You can go as high as 200kΩ, but the resulting sampling rate of 3.2kHz is poor, giving an audio bandwidth of just 1.3kHz.

Although the default rate of 6.4kHz is good enough for voice, after some experimentation, we settled on 82kΩ as the best compromise, giving a sampling rate of 8kHz and around 8.5 seconds of playback time.

While the 100kΩ resistor is an SMD type, since you will probably want to lower the value if you're changing it, you can simply solder another resistor across it.

For example, connecting a 390kΩ resistor across the existing 100kΩ resistor will get you close to the 82kΩ ideal value. You can even connect this resistor via a switch, giving you two different options by merely flicking it.

Note though that if you record with the switch in one position and play back in the other, you will either sound like a chipmunk or Barry White!

While we mounted the switch and resistor inside the Jiffy box, this may be regarded as superfluous – once you've decided on the resistor you require (if any), it could be soldered across R2 and the switch could be left out.

Building it

Once you have gathered the items in the parts list, building it is easy. Solder the bare ends of the supplied lead to the speaker (if they aren't already connected) and then plug this into the header on the module.

Wire up the 2×1.5V battery holder to pins 1 and 3 of CON1, with the positive end to pin 1 (don't get it the wrong way around or you might let the smoke out...)

You can do this quite easily by cutting a female-female jumper lead in half, stripping and soldering the bare ends to the battery terminals, then plugging these into CON1, taking care that the right leads go to the right pins.

You can use a similar technique to wire up the microswitch between pins 2 and 9 of CON1. Alternatively, as we did, you solder the microswitch wires to the appropriate pads on the back of the PCB (either method is fine).

Next, drill the holes in the Jiffy box to accommodate the speaker and the microswitch. Once again, exact positioning is not needed.

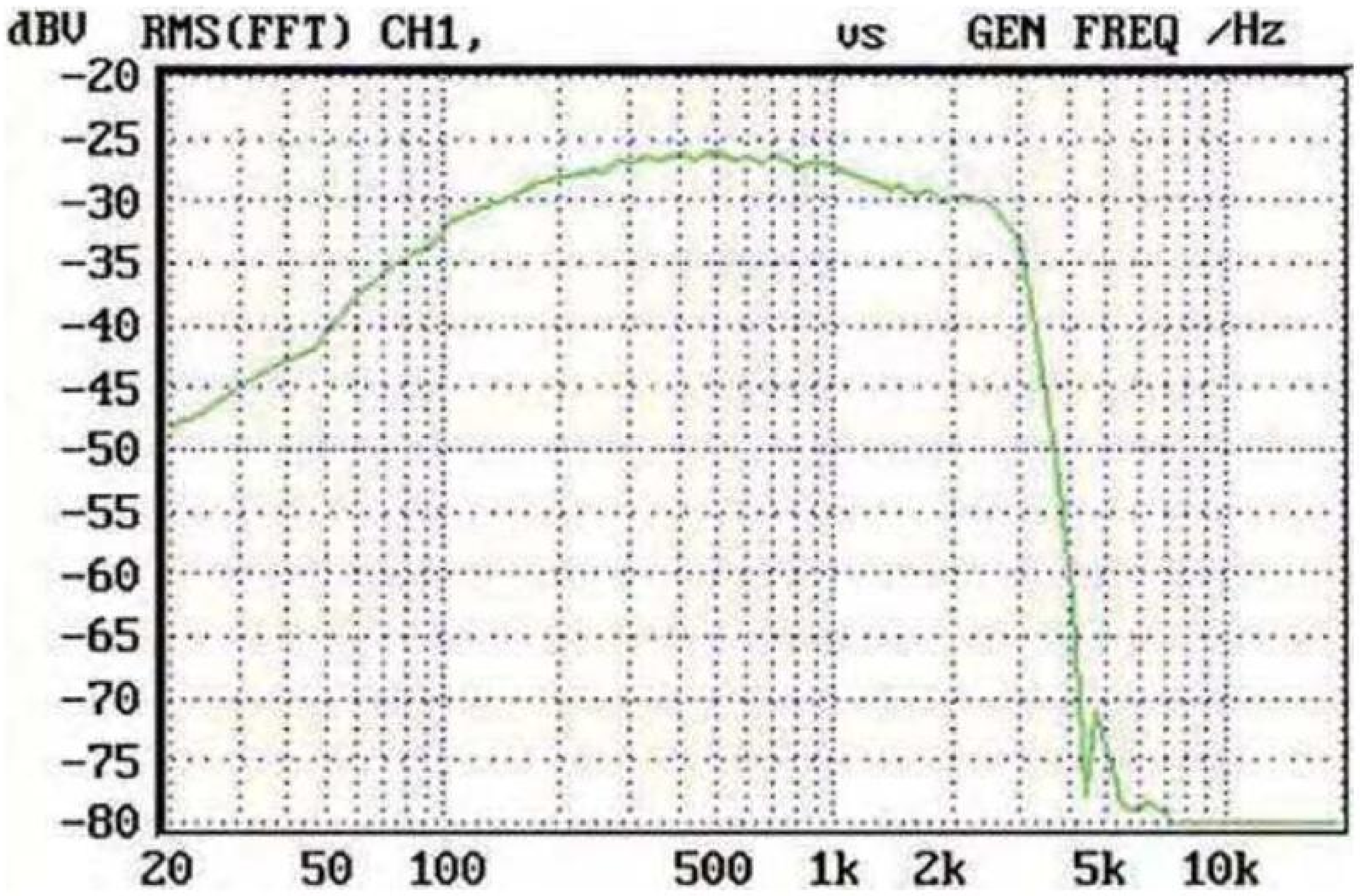


Fig.4: the measured frequency response of the unit from microphone to speaker when the recommended 390kΩ resistor is connected across the 100kΩ onboard resistor from ROSC to ground. This gives a sampling rate of around 8kHz and an audio bandwidth of just over 3kHz. The Nyquist limit (ie, highest possible frequency) when sampling at 8kHz is 4kHz, but the filter's transition band reduces the usable bandwidth to around 3/4 of that figure. This gives eight seconds of playback time and we deem the audio quality to be adequate.

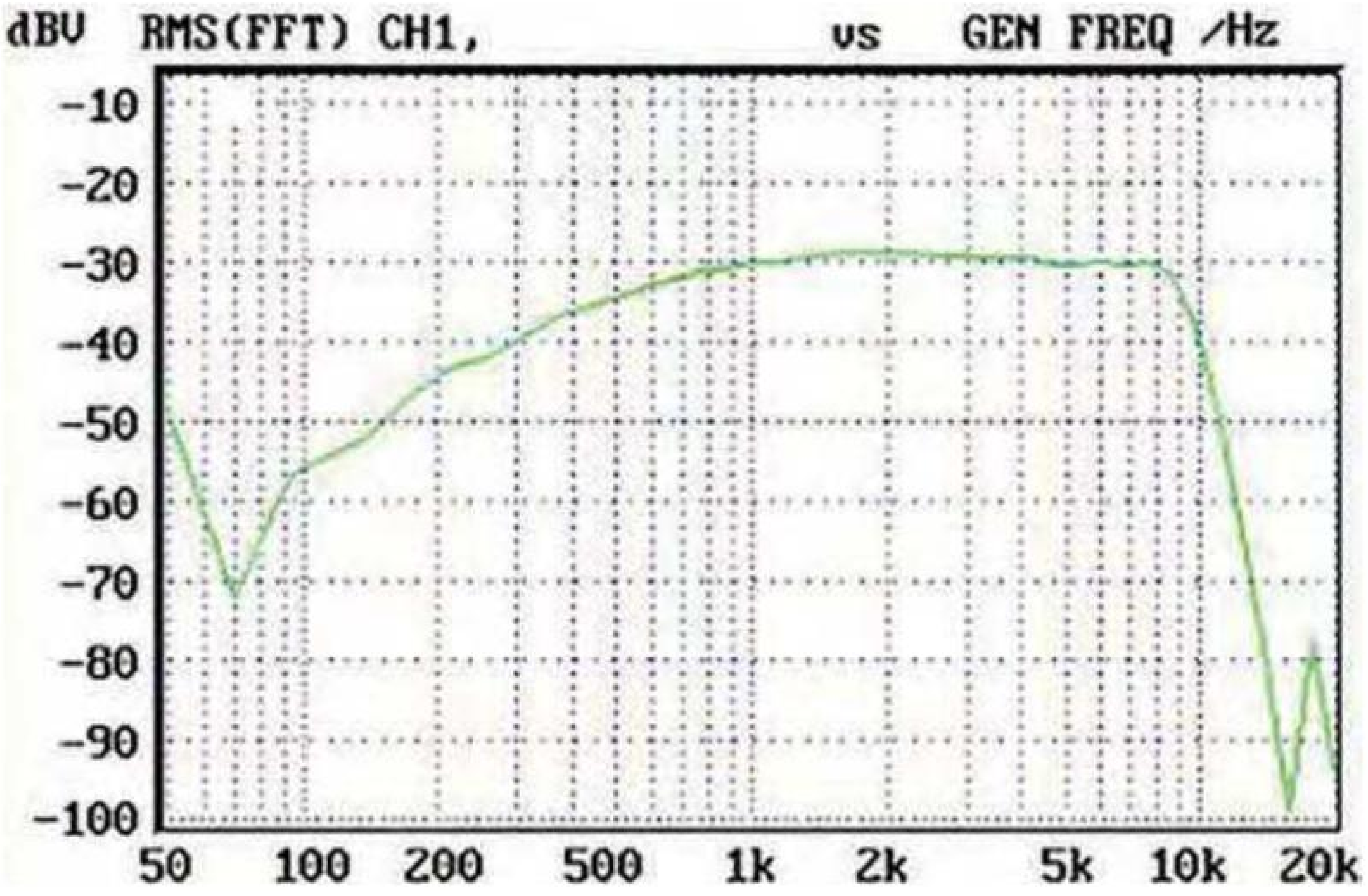


Fig.5: if you solder a 33kΩ resistor in parallel with the existing 100kΩ resistor, you get 25kΩ and that sets the sampling rate to around 20kHz, resulting in the nearly 10kHz of audio bandwidth shown here. The sound quality is better, but the playback time is now limited to around three and a half seconds. That may or may not be enough, depending on what message you intend to convey!

For the speaker, we cut the hole using a 35mm holesaw.The microswitch depends on which type and size you have. Ours (13 × 6mm) had three pins emerging and we drilled three 2mm holes through the lid for these pins.

You'll also want to drill three holes around the periphery of the speaker mounting hole, for machine screws to hold it into place. We drilled three 3mm holes about 3mm out from the edge of the speaker hole, 120° apart.

With these holes just outside where the speaker surround will sit, machine screws with flat washers and nuts will clamp the speaker onto the lid from the inside. See the photo.

If using a switch to control audio quality/recording time (as we did), also drill a hole and mount this now. Put this on one side or end of the Jiffy box – you don't want it to interfere with the microswitch operation.

Depending on the type of battery holder you use, you may need to make a small clamp to hold it in position with a hole drilled in the base of the box. With the holder we used, there is no need to clamp it – it slides down between the PCB guides in the side of the case and locks nicely in place.

Check it twice!

Check that everything is working and record your message. Make sure you are happy with how it sounds, then use neutral-cure silicone sealant to seal the gaps around the edge of the speaker and microswitch holes, and any other holes you've made in the case.

While a Jiffy box is not waterproof, (especially with a speaker in the lid) if you fit the lid on tight, it should survive the sort of splashes it's likely to be exposed to in a mailbox. If you want to be sure, you can always apply silicone around the edges of the lid before attaching it to the case.

All that's left is to place the unit in your mailbox with the microswitch facing up so that anything landing on top of it will trigger the recorded message.

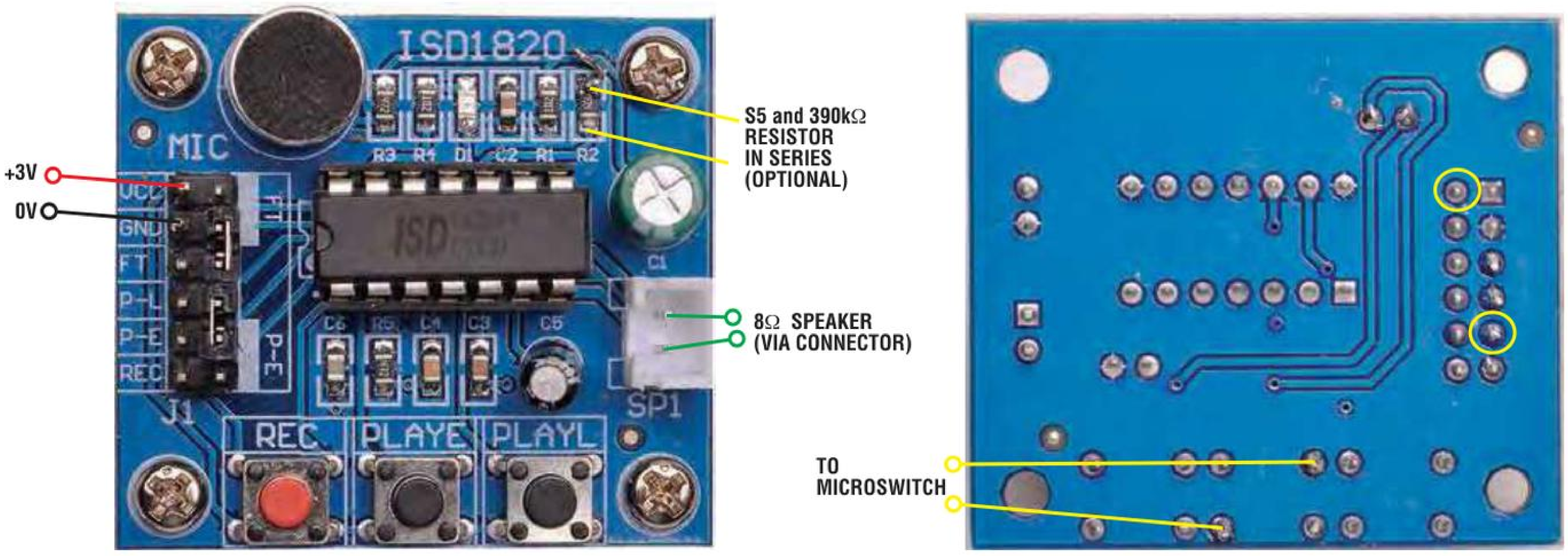
Go ahead, try it out! Then hide behind a tree and wait for an unsuspecting junk peddler to wander by!

And as we mentioned earlier, this project has plenty of other uses; for example – how about a nice pithy message when someone opens up your school bag?

Don't forget that most microswitches can operate in a 'N-O' mode when

PARTS LIST
JUNK MAIL REPELLER

- 1 ISD1820-based voice recorder module with a small speaker and speaker wires
- 1 microswitch
- 1 UB3 Jiffy box (eg, Jaycar Cat HB6023 or Altronics Cat H0153)
- 1 2xAA or 2xAAA cell holder
- 1 390kΩ 1/4W 5% resistor (other values can be used; see text)
- 3 M3 x 10mm panhead machine screws, flat washers and nuts (for mounting the speaker)
- 1 SPST toggle switch (optional, for switchable sound quality)
- 2 female-female or 4 male-female jumper leads
- light-duty hookup wire
- neutral-cure silicone sealant


ISD1820
MIC
+3V
0V
VCC
GND
FT
P-L
P-E
REC
J1
S5 and 390kΩ RESISTOR IN SERIES (OPTIONAL)
8Ω SPEAKER (VIA CONNECTOR)
SP1
REC
PLAYE
PLAYL
TO MICROSWITCH

Front and rear shots of the PCB showing the modifications we made. The connections to the PLAYE switch on the back of the board could also be made on pads 2 and 9 of CON1 (indicated) or indeed to the pins themselves on the top side.

Three 3mm screws, with washers and nuts, hold the speaker in place.

held down and close when released – eg, when a bag is opened!

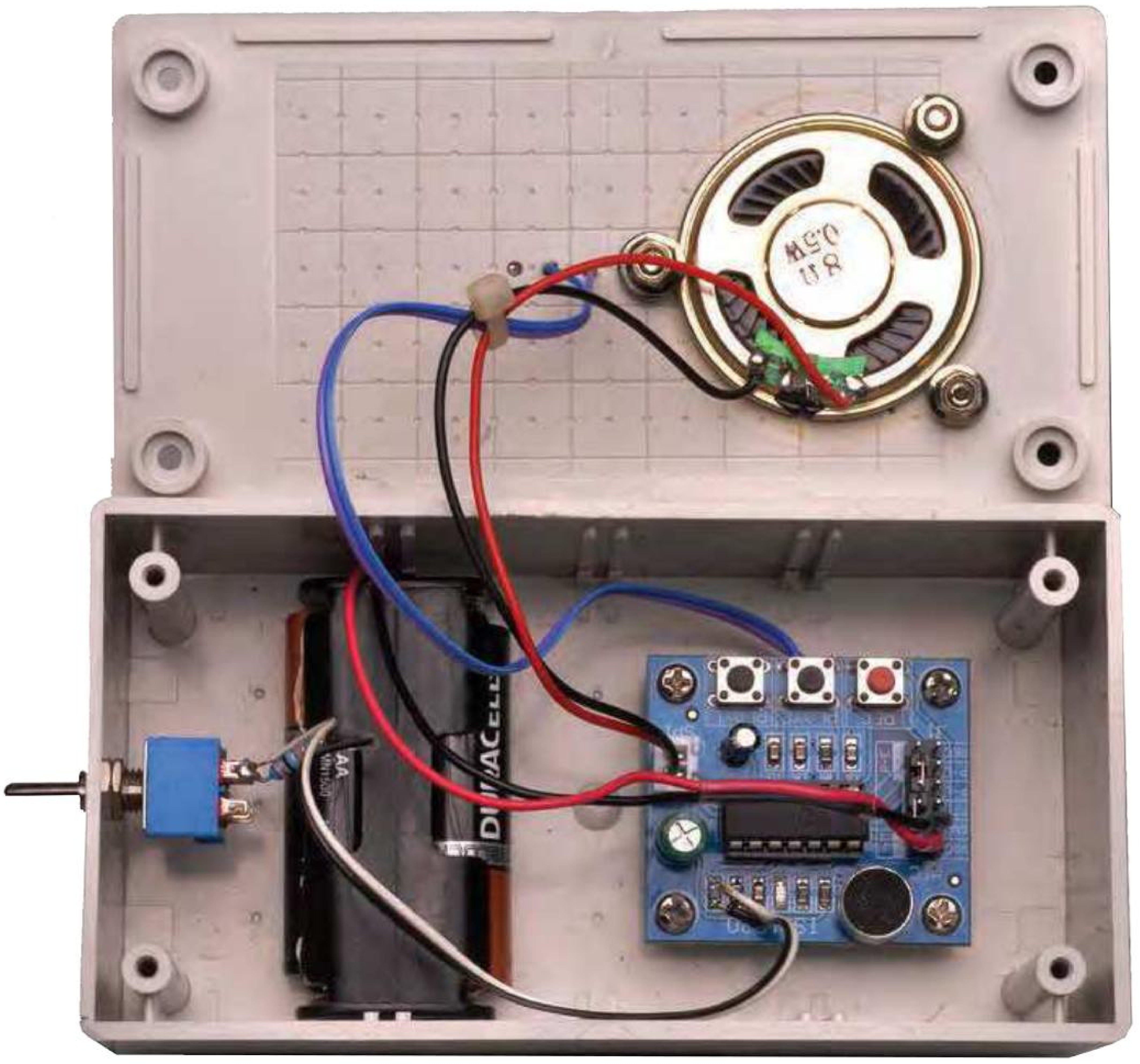
How loud is it?

On the workbench, the answer is 'not very'. Certainly loud enough to be really annoying – but when you place the project in your letterbox, with all its resonances, it's surprisingly loud.

Sure, it's not enough to scare the deliverer into a quivering mess but it should be loud enough for them to hear!

Speaking of placing it in the letterbox, make sure it is placed so that any junk mail (usually larger than legit mail!) can trigger the microswitch but ordinary mail might not have either enough weight or be in the right place to switch it.

Reproduced by arrangement with
SILICON CHIP magazine 2020.
www.siliconchip.com.au



Here is the completed project, ready to scare off any junk mail deliverer. The AA battery holder we used is a nice friction fit in the UB3 Jiffy Box. And the switch at the end (S5) is optional – in fact, we probably wouldn't bother fitting it once we'd decided on the length and quality of our voice recording.

ESR Electronic Components Ltd

All of our stock is RoHS compliant and CE approved. Visit our well stocked shop for all of your requirements or order on-line.

We can help and advise with your enquiry, from design to construction.

3D Printing • Cable • CCTV • Connectors • Components • Enclosures • Fans • Fuses • Hardware • Lamps • LED's • Leads • Loudspeakers • Panel Meters • PCB Production • Power Supplies • Relays • Resistors • Semiconductors • Soldering Irons • Switches • Test Equipment • Transformers and so much more…

Monday to Friday 08:30 - 17.00, Saturday 08:30 - 15:30

**Station Road
Cullercoats
North Shields
Tyne & Wear
NE30 4PQ**

Tel: 0191 2514363 sales@esr.co.uk www.esr.co.uk

Tag-Connect

JTAG Connector Plugs Directly into PCB!!

No Header! No Brainer!

Our patented range of Plug-of-Nails™ spring-pin cables plug directly into a tiny footprint of pads and locating holes in your PCB, eliminating the need for a mating header. Save Cost & Space on Every PCB!!

Solutions for: PIC . dsPIC . ARM . MSP430 . Atmel . Generic JTAG . Altera Xilinx . BDM . C2000 . SPY-BI-WIRE . SPI / IIC . Altium Mini-HDMI . & More

www.PlugOfNails.com

Tag-Connector footprints as small as 0.02 sq. inch (0.13 sq cm)

Practical Electronics

BACK ISSUES

EVERYDAY PRACTICAL ELECTRONICS

AUGUST 2019

PROJECTS • Brainwave Monitor • Super Digital Sound Effects Module – Part 1 • Control your PC with an infrared remote • Watchdog Alarm

FEATURES • The Fox Report • Techno Talk • Net Work • Electronics for a car dynamometer • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans • Electronic Building Blocks

SEPTEMBER 2019

PROJECTS • Intelligent Touchscreen Lathe Speed Controller • Twin Dipole Guitar Speaker • Cheap Asian Electronic Modules – Part 19 • Super Digital Sound Effects Module – Part 2 • White Noise Generator

FEATURES • Techno Talk • Net Work • Circuit Surgery • Audio Out • Practically Speaking • Make it with Micromite • Max's Cool Beans • Electronic Building Blocks

OCTOBER 2019

PROJECTS • Programmable GPS-synced Frequency Reference – Part 1 • Digital Command Control Programmer for Decoders • Opto-isolated Mains Relay

FEATURES • The Fox Report • Techno Talk • Net Work • Using Stepper Motors • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans • Electronic Building Blocks

NOVEMBER 2019

PROJECTS • Programmable GPS-synced Frequency Reference – Part 2 • Cheap Asian Electronic Modules – Part 20 • Tinnitus & Insomnia Killer • Colour Maximite Computer – Part 1

FEATURES • Techno Talk • Net Work • Using Stepper Motors • Circuit Surgery • PIC n' Mix • Audio Out • Make it with Micromite • Max's Cool Beans • Electronic Building Blocks

DECEMBER 2019

PROJECTS • Extremely Sensitive Magnetometer • Useless Box! • Four-channel High-current DC Fan and Pump Controller • Colour Maximite Computer – Part 2

FEATURES • The Fox Report • Techno Talk • Net Work • Circuit Surgery • Using Stepper Motors • Audio Out • Make it with Micromite • Max's Cool Beans • Electronic Building Blocks

JANUARY 2020

PROJECTS • Audio DSP – Part 1 • Isolated Serial Link • Four-channel High-current DC Fan and Pump Controller – Part 2 • Colour Maximite Computer – Part 3

FEATURES • The Fox Report • Techno Talk • Net Work • PIC n' Mix • Using Stepper Motors • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans • Electronic Building Blocks

FEBRUARY 2020

PROJECTS • Audio DSP – Part 2 • Motion-Triggered 12V Switch • USB Keyboard and Mouse Adaptor for Micros • Using Cheap Asian Electronic Modules – Part 21 • Colour Maximite Computer – Part 4

FEATURES • The Fox Report • Techno Talk • Net Work • Practically Speaking • Using Stepper Motors • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans • Electronic Building Blocks

MARCH 2020

PROJECTS • Diode Curve Plotter • Audio DSP – Part 3 • Steam Train Whistle / Diesel Horn Sound Generator • Using Cheap Asian Electronic Modules – Part 22

FEATURES • The Fox Report • Techno Talk • Net Work • PIC n' Mix • Circuit Surgery • Audio Out • Make it with Micromite • Visual programming with XOD • Max's Cool Beans • Electronic Building Blocks

APRIL 2020

PROJECTS • Ultra-low-distortion Preamplifier with Tone Controls – Part 1 • iCEstick – Part 1 • Flip-dot Display

FEATURES • Techno Talk • Net Work • Practically Speaking • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans • Visual programming with XOD

MAY 2020

PROJECTS • 433MHz Wireless Data Range Extender • Bridge-mode Audio Amplifier Adaptor • iCEstick – Part 2 • Ultra-low-distortion Preamplifier with Tone Controls – Part 2

FEATURES • The Fox Report • Techno Talk • Net Work • PIC n' Mix • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans • Visual programming with XOD

JUNE 2020

PROJECTS • AM/FM/CW Scanning HF/VHF RF Signal Generator – Part 1 • Low-cost 3.5-inch touchscreen for the Arduino or Micromite • Ultra-low-distortion Preamplifier with Tone Controls – Part 3

FEATURES • Techno Talk • Net Work • Practically Speaking • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans

JULY 2020

PROJECTS • Speech Synthesiser with the Raspberry Pi Zero • AD584 Precision Voltage References • AM/FM/CW Scanning HF/VHF RF Signal Generator – Part 2 • High-current Solid-state 12V Battery Isolator

FEATURES • Techno Talk • Net Work • PIC n' Mix • Circuit Surgery • Audio Out • Make it with Micromite • Max's Cool Beans • Electronic Building Blocks

BACK ISSUES – ONLY £5.80

£5.80 per issue for UK incl p&p ■ £6.75 Europe Air Mail ■ £7.50 ROW Air Mail

We can supply back issues of PE/EPE by post.

We stock magazines back to 2006, except for the following:

2006 Jan, Feb, Mar, Apr, May, Jul
2007 Jun, Jul, Aug
2008 Aug, Nov, Dec
2009 Jan, Mar, Apr
2010 May, Jun, Jul, Aug, Oct, Nov
2011 Jan
2014 Jan
2018 Jan, Nov, Dec
2019 Jan, Feb, May, June

Issues from Jan 1999 are available on CD-ROM / DVD-ROM

If we do not have a a paper version of a particular issue, then a PDF can be supplied for the same price – your email address must be included on your order.

Please make sure all components are still available before commencing any project from a back-dated issue.

ORDER FORM – BACK ISSUES

☐ Back issues required (month / year) . . . .

Name . . . .

Address. . . . .

. . . . .Tel: . . . .

Email . . . .

☐ I enclose cheque/PO to the value of £ . . . .

☐ Please charge my Visa/Mastercard £ . . . .

Card No . . . .

Valid From . . . . Card Expiry Date . . . .

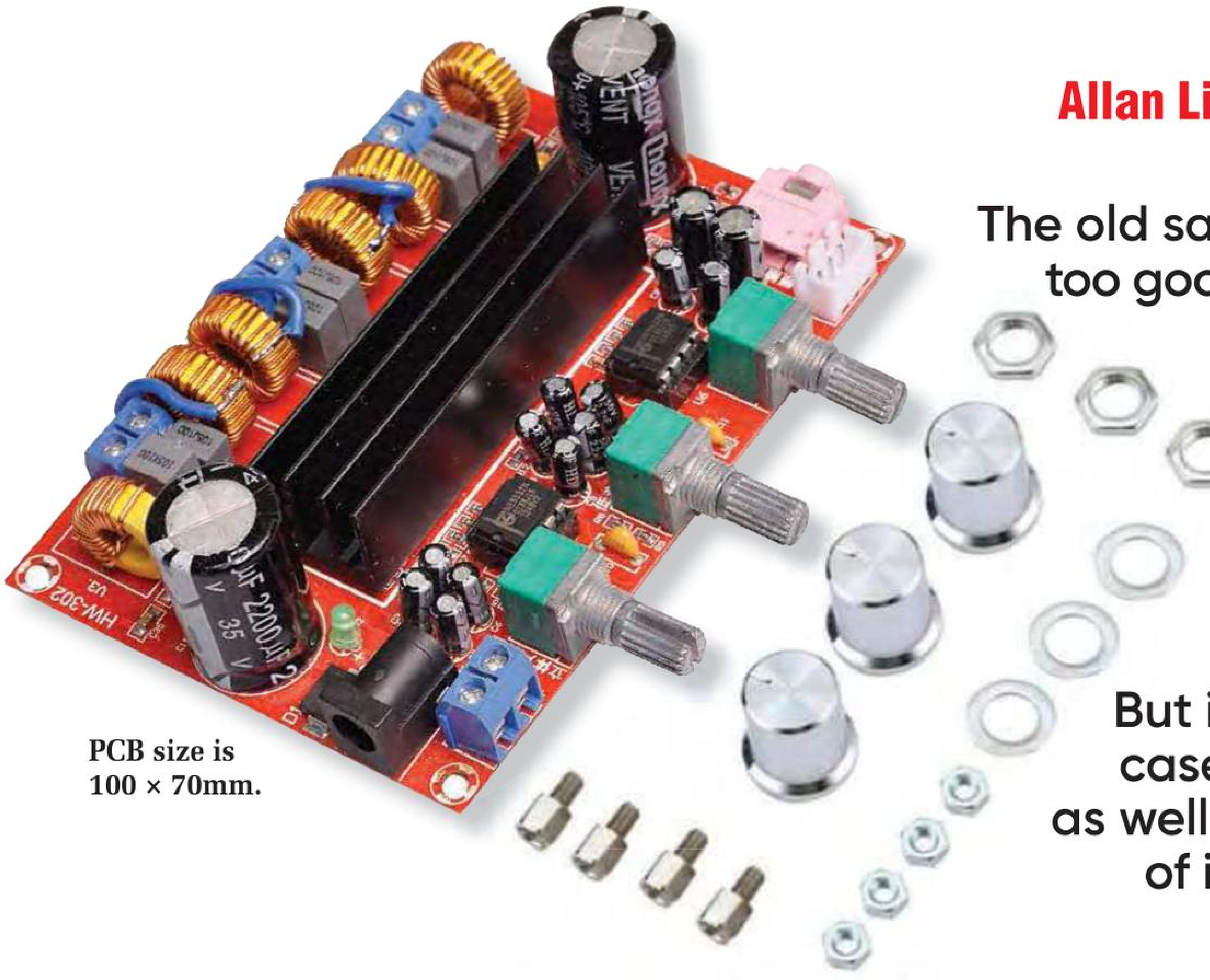
Card Security Code . . . . (Last three digits on or under the signature strip)

SEND TO: **Practical Electronics, Wimborne Publishing Ltd**
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU.
Tel: 01202 880299 **Fax:** 01202 843233.
Email: stewart.kearn@wimborne.co.uk **On-line Shop**: www.epemag.com

Payments must be in £ sterling – cheque drawn on a UK bank. ALl items are normally posted within seven days of receipt of order. Send a copy of this form, or order by letter / online if you do not wish to cut your issue.

Bargain Modules
Class-D Stereo Plus Subwoofer Amplifier

By
Allan Linton-Smith



PCB size is 100 × 70mm.

The old saying says that 'if it sounds too good to be true, it probably is'. So if we told you that you could get an assembled 3 × 50W amplifier module for around £7.50, you would probably be thinking that it would be a load of junk. But in this circuit, that isn't the case! This one works almost (!) as well as advertised – and most of its shortcomings are easily be addressed.

The Class-D 3 × 50W amplifier module (stereo plus subwoofer) shown above can be purchased (at time of going to press) for about £7.50 from AliExpress (or less – do google it!).

For a bit more money, you can get the 5x50W amplifier module with built-in Bluetooth support shown opposite.

Both run from 5-27V DC, provide decent performance and appear to be very good value for money.

The XD172700 module

The module above uses the latest power IC from Texas Instruments (TI), the TPA3116D2 IC (2017 revision G), who describe it as a '15W, 30W, 50W Filter-Free Class-D Stereo Amplifier Family With AM Avoidance'.

The chip measures just 11 × 6.2mm. Two are used on the first board: one is used in stereo mode for the left and right channels, and the other in mono (bridged) mode for driving a subwoofer.

These amplifier chips are fed audio by two NE5532 ICs used as preamplifiers and to provide the subwoofer low-pass filter.

You don't have to worry about soldering the SMD TPA3116D2 chips because this has all been done for you! Our suggested modifications require a little bit of much simpler soldering.

The board comes with everything, even the heatsink, which is shared by both amplifier ICs. It even came with a set of standoffs, nuts and bolts for mounting it in a chassis, plus shiny knobs for the pots! All you need to do is wire up the power supply, audio input and speaker output terminals.

The board has two audio input options: you can use either the 3.5mm stereo jack socket or a three-pin JST header. And there are two options for power supply; either a PCB screw terminal or a 5.5mm DC barrel socket for a plugpack or inline power supply.

The board requires a simple DC supply, and this simplifies things significantly because you can use just about any supply that produces 5-24V DC: an old laptop supply or any other high current source, including a car or electric drill battery. You could even use a 5V USB charger. But to get the full output power, you need around 24V at 6-7A.

Note that to get the full power output you will also need 4Ω speakers. Higher impedance speakers cannot be driven to quite as high power levels. For example, if you use 8Ω speakers, with the appropriate power supply, you will get around 30W maximum from the left and right channels.

The amplifier ICs have a high power supply rejection ratio (PSRR), so you don't need a super-smooth DC supply. It will reject 70dB of ripple, meaning you can have up to 200mV peak-to-peak ripple before you're likely to notice buzz or hum creeping into the audio outputs.

For testing, we used a 24V 7A DC plugpack which cost £18 including postage. 24V × 7A = 168W, so with a 90% claimed peak amplifier efficiency, we should get a total theoretical output of around 150W RMS – ie, around 2 × 38W into 4Ω for the left and right channels and about 75W into a 2Ω subwoofer.

The efficiency of the device varies significantly with supply voltage and output power (see Fig.1). It is typically 40-70% at low power levels, ie, below 5W. If you only require power levels up to 10W into 4Ω speakers you are better off with a 6-12V DC supply because this will give you 70-90% efficiency and it won't cause any overheating problems (see Fig.1).

Your best approach is to decide what power output you need and then choose your power supply to deliver this with the highest efficiency. Otherwise, the device may overheat and automatically shut down during use.

Power output figures

The measured power for this module is good but not quite up to the claim of 2 × 50W + 100W.

During testing, we did manage to get 2 × 50W into 4Ω and 2 × 30W into 8Ω as expected. But we were not able to get the full 100W into 2Ω from the subwoofer output because the device protection circuit sent the output into high impedance and it cut out. We were only able to get about 50W into the sub.

This is no doubt due to poor design of the subwoofer section; we suspect that the IC has not been correctly configured for mono operation. It may be possible to fix this by changing some of the passive components connected to the subwoofer amplifier IC, but we haven't tried that.

So basically, you can expect to get about the same amount of power from the subwoofer channel as you can from the left and right channels, taking into account the possibility that your sub may have a different impedance from the other speakers.

Frequency response

The quoted frequency response by the supplier is 20Hz to 20kHz with no ±decibel figure, which is quite common to see but also a pretty-much useless statement. So we decided to measure the frequency response accurately.

First, we did a listening test which exposed a lack of treble with cymbals, triangles and slightly muffled brass. The measured response, as shown in Fig.2, confirms our subjective impression.

There is a significant drop-off in the output above 1kHz. We did this test at 1W and 5W output levels, using a 12V DC supply for convenience.

So the out-of-the-box response is poor, and you can clearly hear the lack of treble. It's down by 8dB by 20kHz.

A glance at the TI data sheet (**www.ti.com/lit/ds/symlink/tpa3116d2.pdf**) indicates that when properly implemented, the IC's frequency response should be almost flat to about 40kHz.

The data sheet also recommends that the LC filter after the output stage, if fitted, should have a 10μH inductor and 680nF capacitor on each output pin. We measured the supplied LC filter at 55μH and 1μF, which explained the drastic reduction in high-frequency response. We tried reducing the output inductor values to 10μH, which considerably flattened the frequency response.

As per the data sheet, high-current ferrite beads can be used in place of the inductors, if the capacitors are also changed to 1nF.

This will not be as effective at reducing radiated emissions, however, and doing this will require quite a bit of soldering which may damage the dual-layer PCB.

Changing the inductor values has another benefit besides flattening the frequency response; we found that they got hot during use because the wire used is too thin.

Audio inductors should be air-core types to avoid non-linearity in the core material. We published instructions for winding a 2.2μF inductor on page 28 of the February issue. To make a 10μH inductor use 30.5 turns of 1mm diameter enamelled copper wire on standard bobbins available from Jaycar and Altronics.

You then just need to remove the existing inductors and solder the improved ones into place. Keep them as close to the PCB as possible and mount them all with the same orientation to reduce magnetic field interactions.

Ideally, you should replace the 1μF capacitors with 680nF capacitors, as per the data sheet; but in practice it doesn't make that much difference.

You can see the revised frequency respone (after changing the inductor values) as the blue trace in Fig.2

With the 10μH inductors and 1μF capacitors, it shows a slight lift at 20kHz, continuing to rise to 30kHz, then dropping sharply to –60dB at 1MHz.

Naturally, after doing that, the unit sounded much better, with an excellent high-frequency response; very different from our first listening test!

The subwoofer response is also shown in Fig.2. It has a peak at 28Hz and is –20dB at 250Hz, which is close to ideal.

The subwoofer amplifier can put out significant power and the IC is supposed to handle 2Ω speakers, but we found that 4Ω is the minimum for this particular module. In fact, you won't find many 2Ω drivers (outside of cars), anyway.

You may notice that after this modification the module has a slight (2dB) rise at the low-frequency end, close to 20Hz. This is probably due to crosstalk with the subwoofer section and the design of the PCB, but it should not be a problem because most loudspeakers will not respond to such low frequencies. Either way, a small amount of low-end boost will generally improve the response of most loudspeakers.

AM radio frequency avoidance

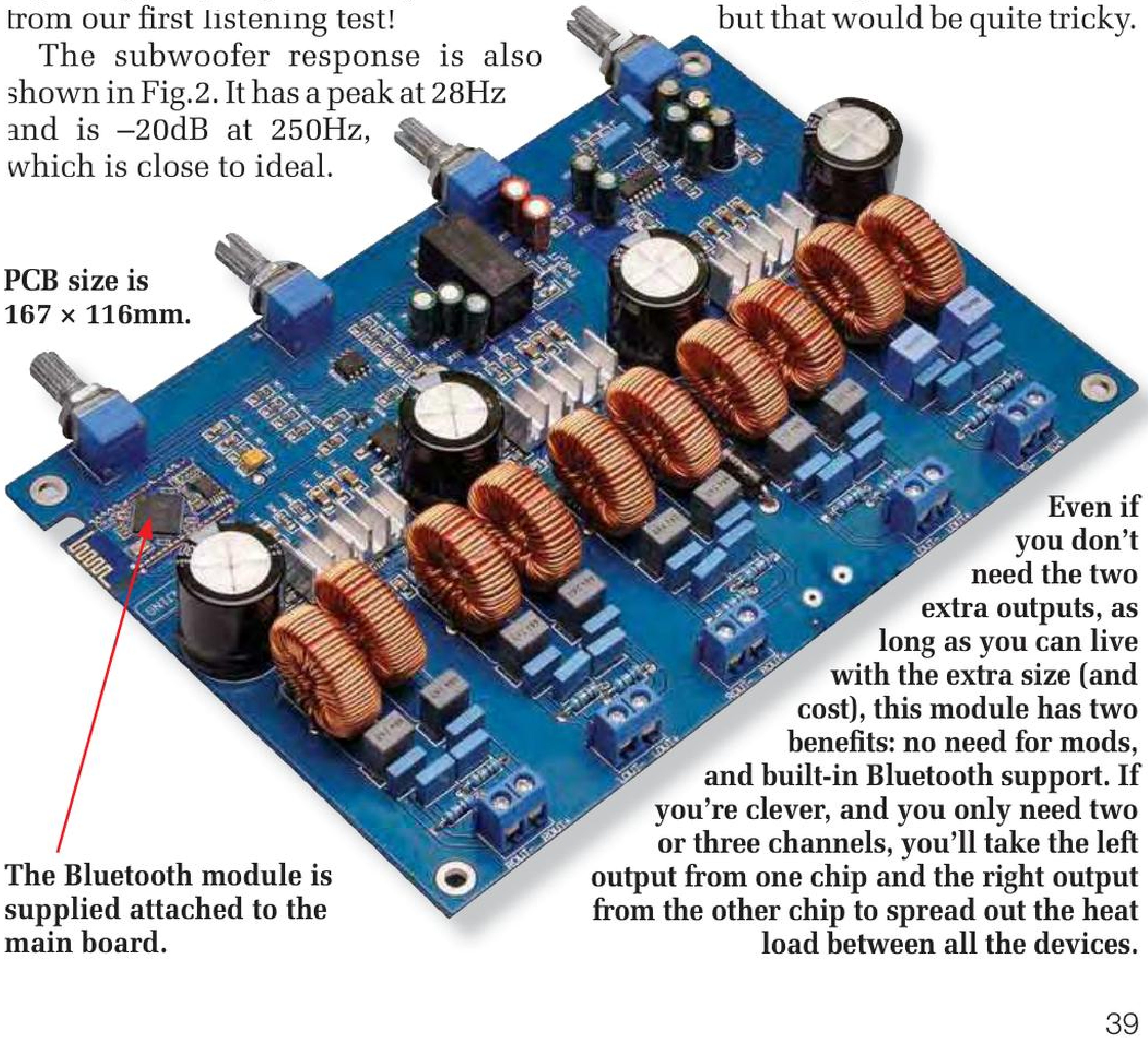
The TPA3116D2 has advanced oscillator/PLL circuitry which employs multiple switching frequency options to avoid AM interference.

These options cover 15 different frequencies, ranging from 376kHz to 1278kHz, so it can be set to avoid the AM band in most countries.

Our module was pre-set at 400kHz (403.5kHz measured) so that only the first harmonic will fall into our local AM band.

We also checked the output with a spectrum analyser and found that the first harmonic (807kHz) was 57dB lower than the audio output signal level, so there should be very little interference with AM radio receivers (see Fig.3).

If you are going to use the module in other places where 400kHz radiation could be a problem, you could modify the unit according to the data sheet, but that would be quite tricky.



PCB size is 167 × 116mm.

The Bluetooth module is supplied attached to the main board.

Even if you don't need the two extra outputs, as long as you can live with the extra size (and cost), this module has two benefits: no need for mods, and built-in Bluetooth support. If you're clever, and you only need two or three channels, you'll take the left output from one chip and the right output from the other chip to spread out the heat load between all the devices.

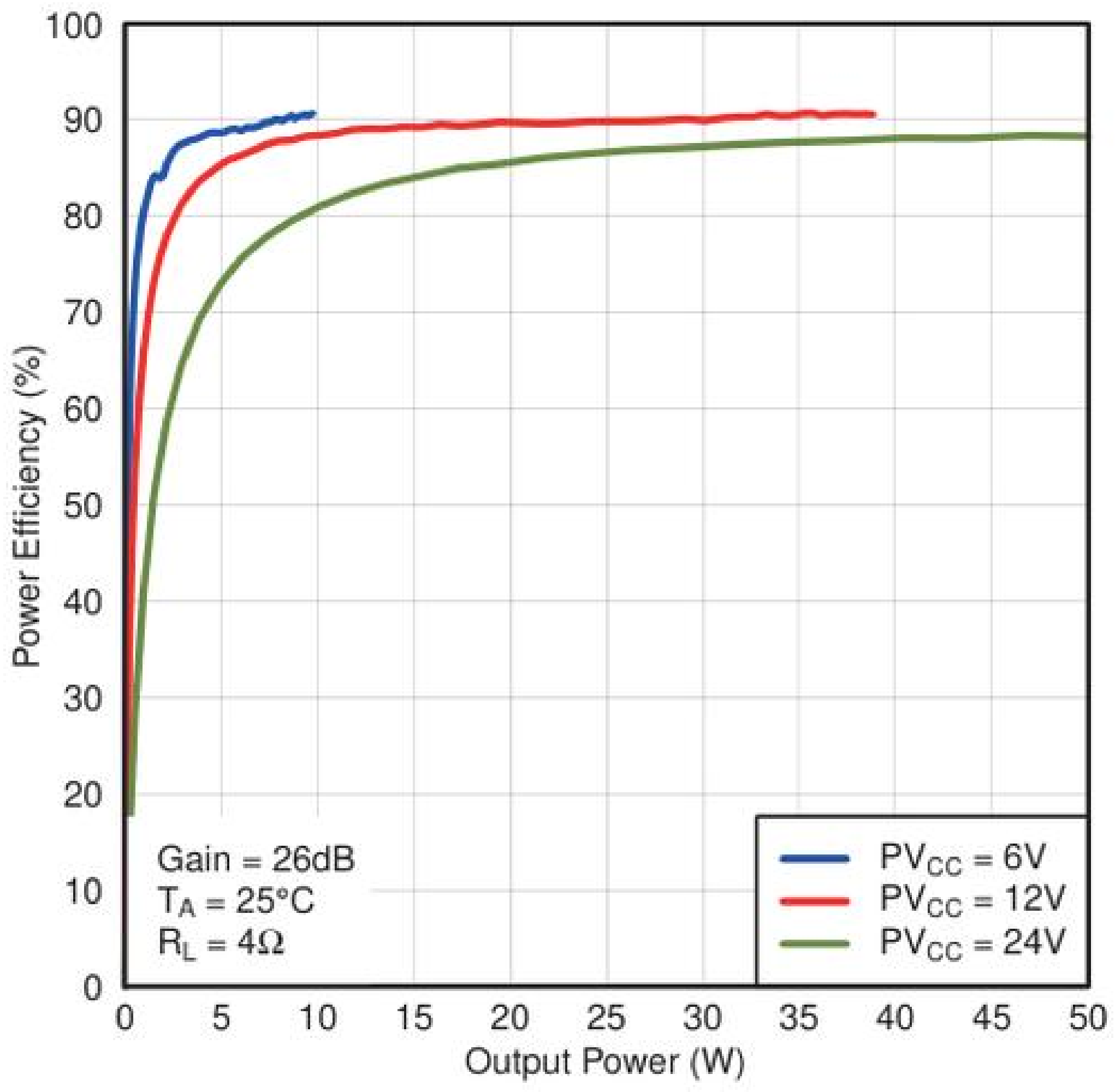


Fig.1: sample efficiency curves from the Texas Instruments TPA3116D2 data sheet. Efficiency is higher with lower supply voltage, but of course, maximum power is also lower in those cases. Efficiency also increases with output power; in other words, device dissipation does not increase much as the output power rises.

So we suggest that instead you try to keep the speaker leads short – less than 1m if possible – so they make for poor transmitting aerials. The spectrum from 500Hz to 40MHz is otherwise very clean.

Distortion and noise (THD+N)

The unit is quoted as having a THD+N figure of 0.1% at 1kHz with a 25W output. We decided to verify this with some measurements.

The maximum power into an 8Ω load is 40W RMS and the THD+N reading was 1% when clipping started to be noticeable at this level. The high THD+N at very low power levels is merely noise. As expected, the module will deliver 50W into 4Ω loads.

Fig.5 shows a plot of THD+N vs frequency for the module. These figures are the best that we were able to achieve after changing the output inductors. The distortion above 10kHz may be higher than indicated because we used a 20kHz 'brick wall' filter to eliminate subharmonics from the 400kHz switching frequency, which otherwise would have affected the measurements.

The 80kHz bandwidth measurements we usually take with linear amplifiers cannot be made with Class-D amplifiers. Therefore, we took some intermodulation distortion (IMD) measurements to clarify the level of distortion at higher frequencies.

The IMD measurements were taken by injecting the SMPTE-standard frequencies of 500Hz and 2kHz (2:1) and the resultant spectrum shows acceptably low noise up to 24kHz. The average level is 0.11% which verifies the THD+N measurements; this is not bad for a Class-D amplifier.

Crosstalk

We checked out the crosstalk of the amplifier module (Fig.5) and the results were as not as good as specified, probably because of the design of the PCB and the interaction of the output inductors, which cause feedback into the opposite stereo channel.

There is not a lot you can do about this; it may be possible to re-locate the inductors or substitute ferrite beads, but if you want really good crosstalk performance, given its low cost, you could simply use a separate module for the left and right channels.

While we were working on this article, similar modules have appeared on eBay for around £3. So it's hardly worth arguing about!

XD172700 Class-D amplifier features and specifications

- 3 × 50W RMS into 4Ω (21V DC supply)
- 3 × 30W RMS into 8Ω (24V DC supply)
- Supply voltage: 4.5-27V DC
- THD+N: typically around 0.05% at 1kHz, 1W
- Frequency response: 20Hz-20kHz, +3,–0dB (after modifications)
- Efficiency: up to 90% (only needs a small heatsink)
- Switching frequency: 400kHz ±3kHz
- Self-protection circuits: over-voltage, under-voltage, over-temperature, DC offset, over-current and short-circuit protection.
- Input connectors: 3.5mm stereo jack socket or 3-pin JST header
- Output connectors: 3 × 2-way terminal blocks
- Power connectors: 2-way terminal block and DC barrel socket
- Module size: 100 × 70 × 30mm

Protection features

The TPA3116D2 is a well-protected device and has self-protection for over-voltage and under-voltage conditions, as well as output DC fault, short-circuit, overload and over-temperature conditions. When an over-current, short-circuit, over-temperature or DC offset fault is detected, the module switches itself off and you need to cycle power to restore its function.

No point changing the op amps

As mentioned earlier, the unit we obtained came with two NE5532 op amps in sockets. Most dual op amps in DIP-8 packages have the same pinout,

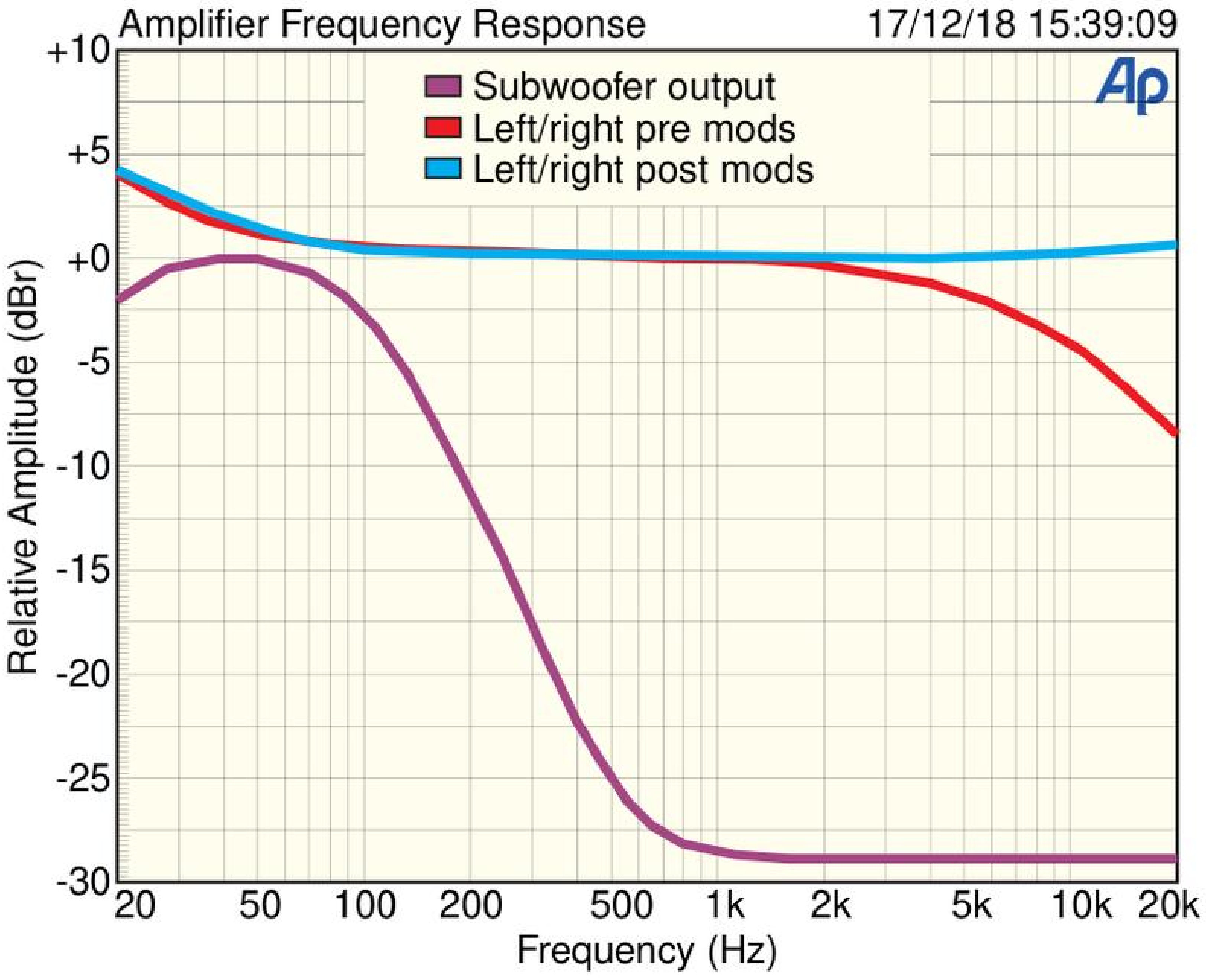


Fig.2: frequency response of the 2+1 channel amplifier module before and after we modified it. The mauve curve shows the subwoofer output, which purposefully rolls off at around 100Hz. The left/right response, as supplied, is in red, and post-mods is in blue. It's now much flatter above 1kHz, and it sounds a lot less muffled.

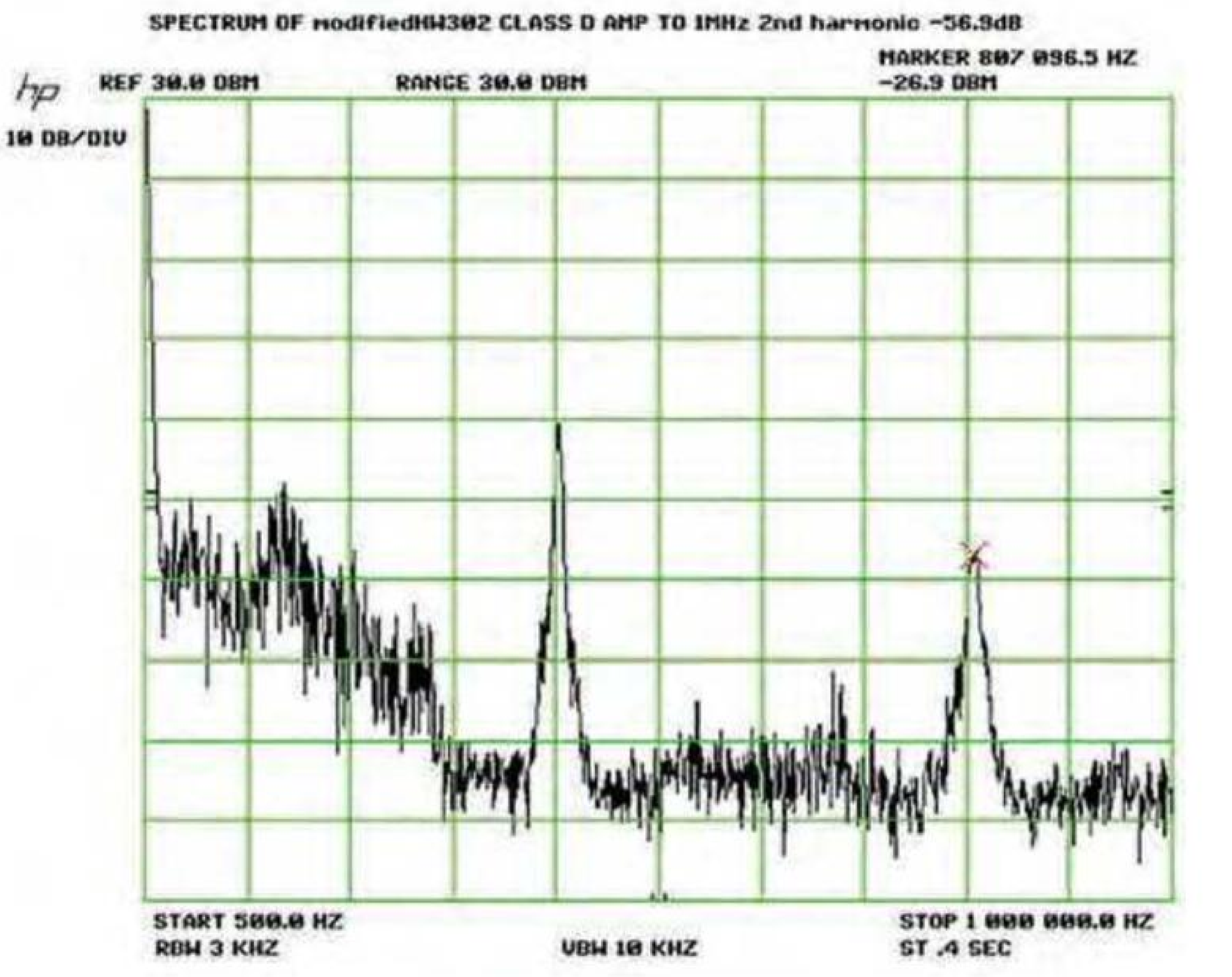


Fig.3: spectrum analysis of the output waveform shows that the main peak at 403kHz, representing what's left of the switching waveform after filtering, is 40dB below the audio signal, while its first harmonic at 806kHz (in the AM broadcast band) is at –57dB, so the amplifier should not cause too much AM interference. Nevertheless, we'd keep the speaker leads as short as possible.

Yuanjing Class-D amplifier features and specifications

- Inputs: 3 separate channels (left, right, subwoofer)
- Outputs: 5 × 50W RMS into 4Ω (21V DC supply) or 5 × 30W RMS into 8Ω (24V DC supply)
- Supply voltage: 4.5-27V DC
- THD+N: typically around 0.05% at 1kHz, 1W
- Frequency response: 20Hz-20kHz, ±1dB
- Efficiency: up to 90% (comes with small heatsinks fitted)
- Switching frequency: 400kHz ±3kHz
- Self-protection circuits: over-voltage, under-voltage, over-temperature, DC offset, over-current and short-circuit protection.
- Input connectors: 3-way pin header or Bluetooth wireless
- Output connectors: 5 × 2-way terminal blocks
- Power connector: solder pads
- Module size: 165 × 115 × 25mm

so it's easy to swap them – but there isn't much point.

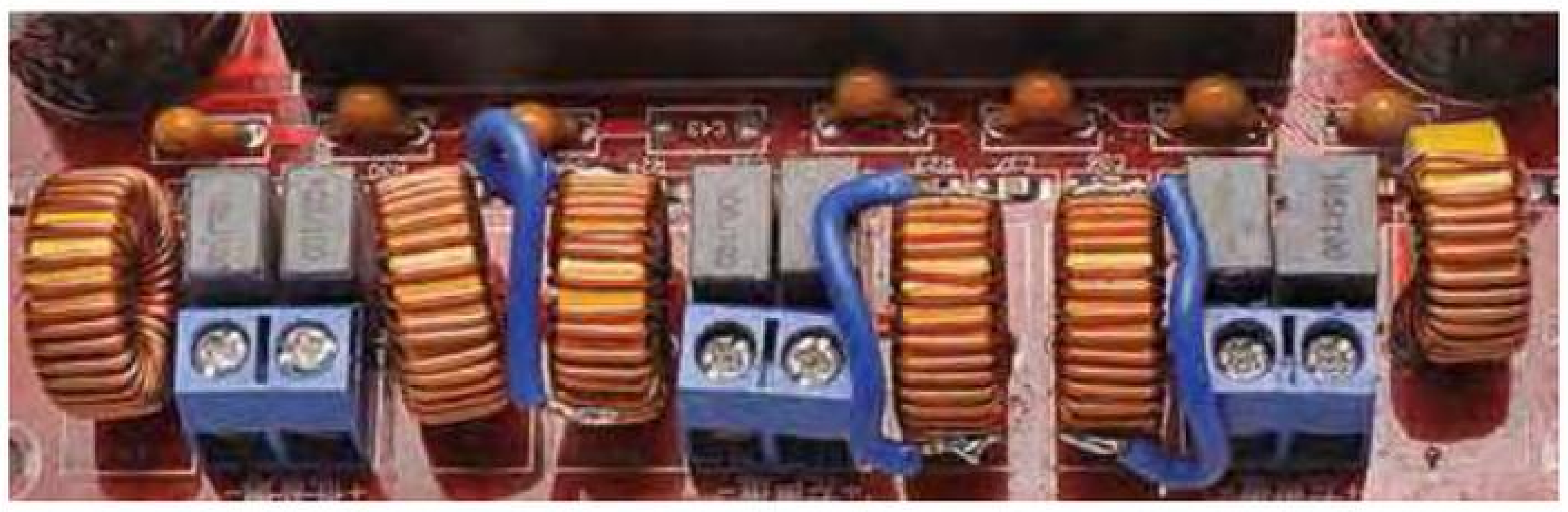
First, while the NE5532 is an old design, it has stood the test of time and even by today's standards still has outstanding performance.

And second, the distortion and noise in this amplifier is dominated by the amplifier ICs themselves and not the op amp-based preamplifiers.

We tried replacing the NE5532 with newer OPA1642s (soldered to SOIC-to-DIP adaptors) but the improvement in performance was so minor as to be insignificant. If you must change the op amps, don't forget to fit them in the right orientation!

Getting one

There are many similar modules available with a different size, layout, components, connectors and so on. You may want to look for one that's visually identical to ours, since it is at least a known quantity. There are many possible sources but here is one to get you started: **www.aliexpress.com/item/32810347968.html**



As noted in the article, the inductors on the 172700 unit had much too high a value to give a good frequency response. Not wanting to spend any money on new inductors (they would cost more than we paid for the whole module) we tried partially unwinding some of them. That worked, but it was a lot of work. So for the remainder, we shorted out 15 turns by soldering thin wires in place (after scraping off the enamel insulation from the wire), as seen here. This dropped their inductance down to roughly the right value.

The Yuanjing module

Since we noticed so many other similar modules were available, we decided to try a second one; specifically, one with built-in Bluetooth support.

The one we've chosen has no obvious model number, but since it has 'Yuanjing' written in copper tracks in the corner near the Bluetooth module, and this is presumably the manufacturer, that's how we're referring to it.

You can find this module for sale at prices from about £17.50 to £25 on eBay and AliExpress, although the latter has a better selection. Search for 'tpa3116 4.1' and look for a blue PCB matching the one shown in this article. This one appears to be the best deal at the time of writing: **www.aliexpress.com/item/32799510099.html**

This module may be for motor vehicles given that it has two pairs of essentially identical left/right outputs – perhaps to drive front/rear car speakers. The four pots along the front control overall volume, subwoofer volume and front and rear volume independently.

Even if you don't need the extra channels, there are two big advantages to this module. One, we didn't need to make any modifications to get good performance out of it; it appears to have the correct output filter components from the factory. And two, the built-in Bluetooth audio receiver is very handy for wirelessly playing audio from a mobile phone or tablet.

It works seamlessly. When a Bluetooth device is connected, it switches a relay to divert the Bluetooth audio to the amplifier chips. With no Bluetooth connected, audio comes in via a three-way pin header. The subwoofer signal is generated by mixing the left and right channel signals and then feeding it through a low-pass filter.

Like the XD172700, the subwoofer output on this module does not appear capable of the claimed 100W. We think that in both cases, they simply have not wired up the IC correctly for BTL operation. It's merely using one of the two available channels and so is only capable of driving 4-8Ω loads to the

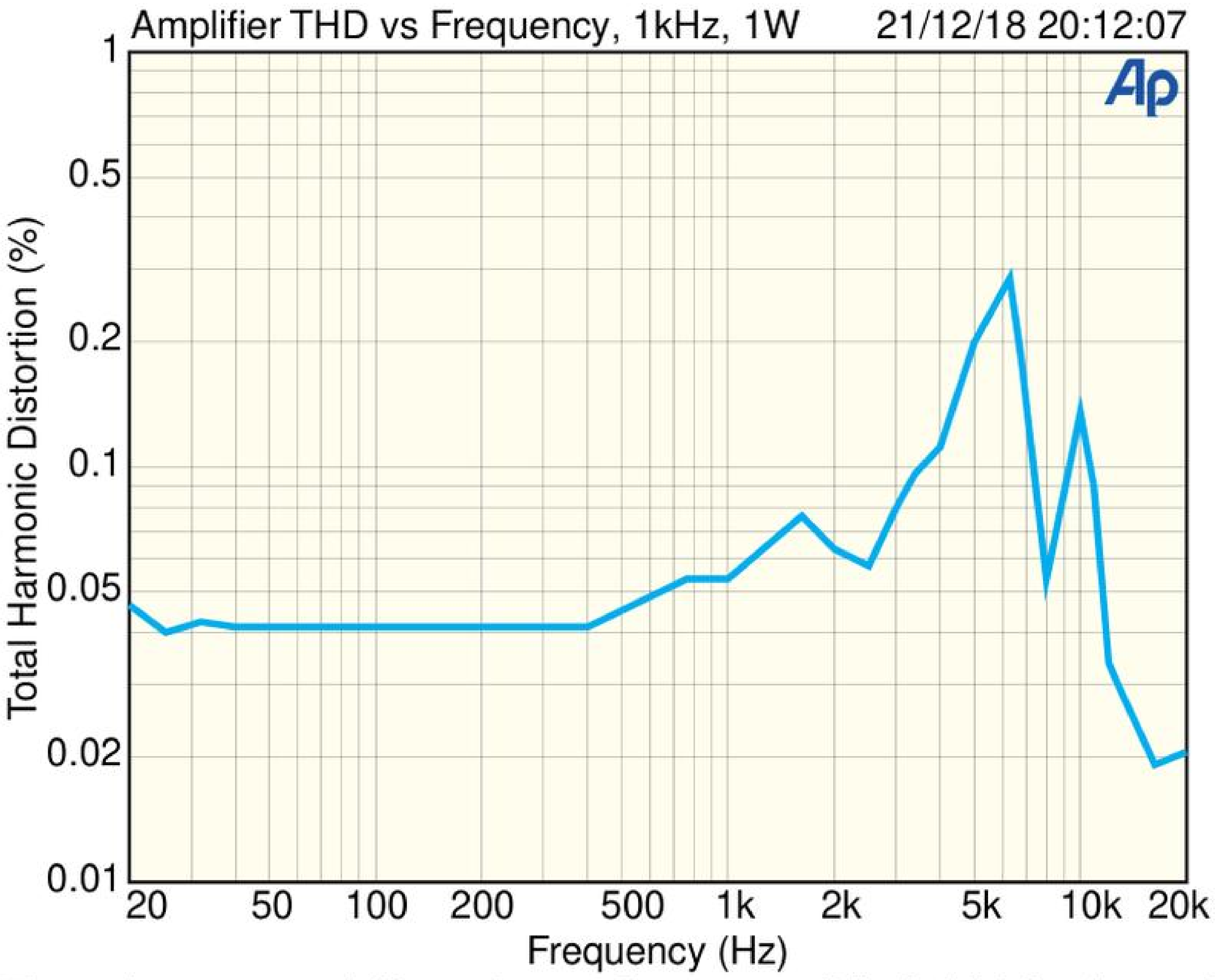


Fig.4: the measured distortion performance of the left/right channels on our sample module (after output filters mods), into an 8Ω resistive load. While not quite as good as the amplifier designs we publish, it's below 0.1% THD+N up to about 3.5kHz (with a 20kHz bandwidth) which is not too bad. It certainly sounds acceptable. We use a 20kHz filter to remove the switching residuals, hence the drop in readings above 6kHz, above which the main harmonics are filtered out.

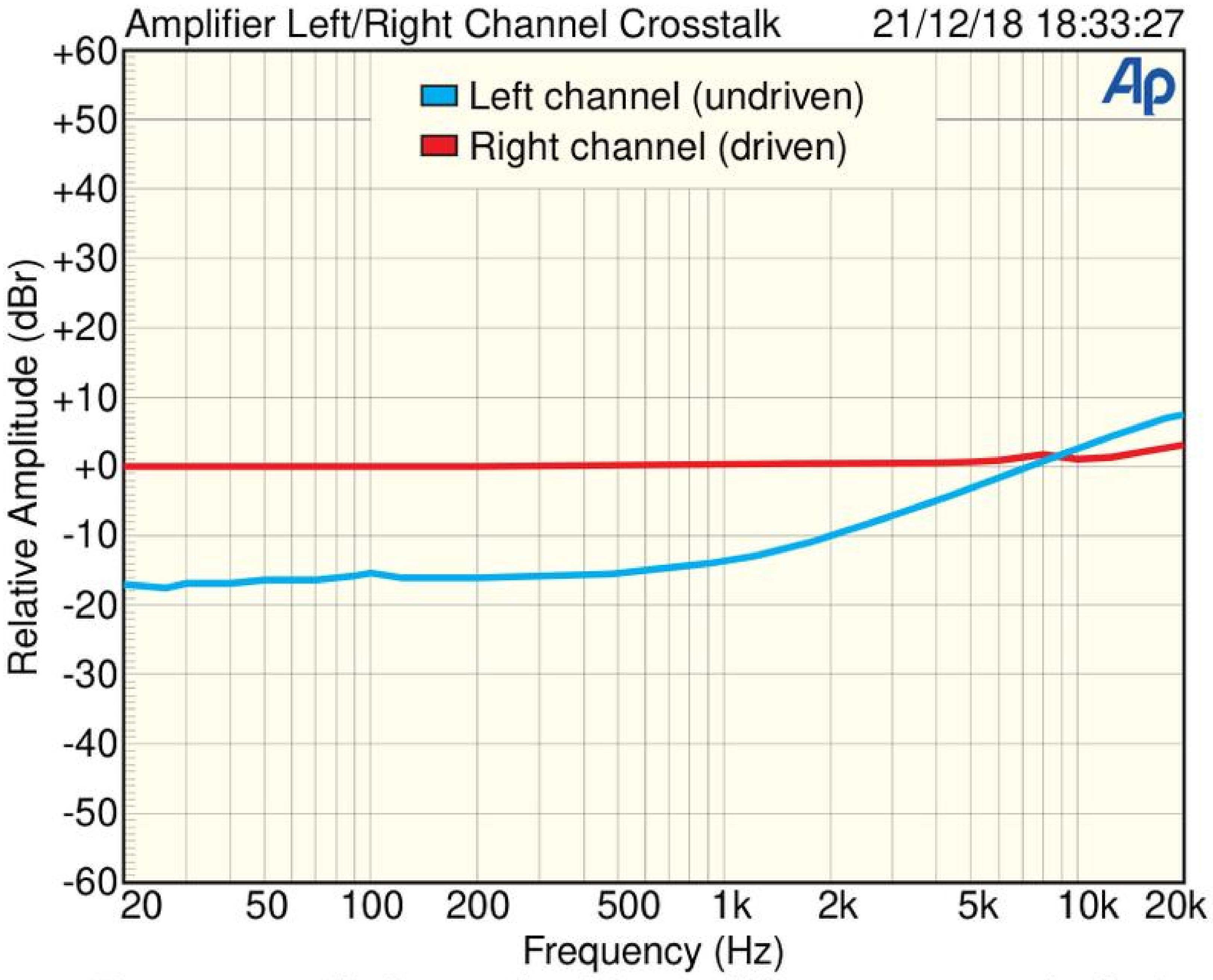


Fig.5: crosstalk figures for this amplifier are not particularly great, with less than 20dB separation between channels. This is probably due to the close proximity of the output filter inductors for each channel. This generally isn't a problem when playing regular music recordings, but if it bothers you, you have the option of using two separate modules, one for each stereo channel.

Fault Reporting

FAULT	TRIGGERING CONDITION (typical value)	FAULTZ	ACTION	LATCHED/SELF-CLEARING
Over Current	Output short or short to PVCC or GND	Low	Output high impedance	Latched
Over Temperature	T_j >150°C	Low	Output high impedance	Latched
Too High DC Offset	DC output voltage	Low	Output high impedance	Latched
Under Voltage on PVCC	PVCC < 4.5V	–	Output high impedance	Self-clearing
Over Voltage on PVCC	PVCC > 27V	–	Output high impedance	Self-clearing

Fig.6: the self-protection features of the TPA3116D2 IC.

same power levels as the left and right channels. But still, overall, the performance isn't bad, especially considering the price and the convenience of running off a single, relatively low voltage DC supply rail.

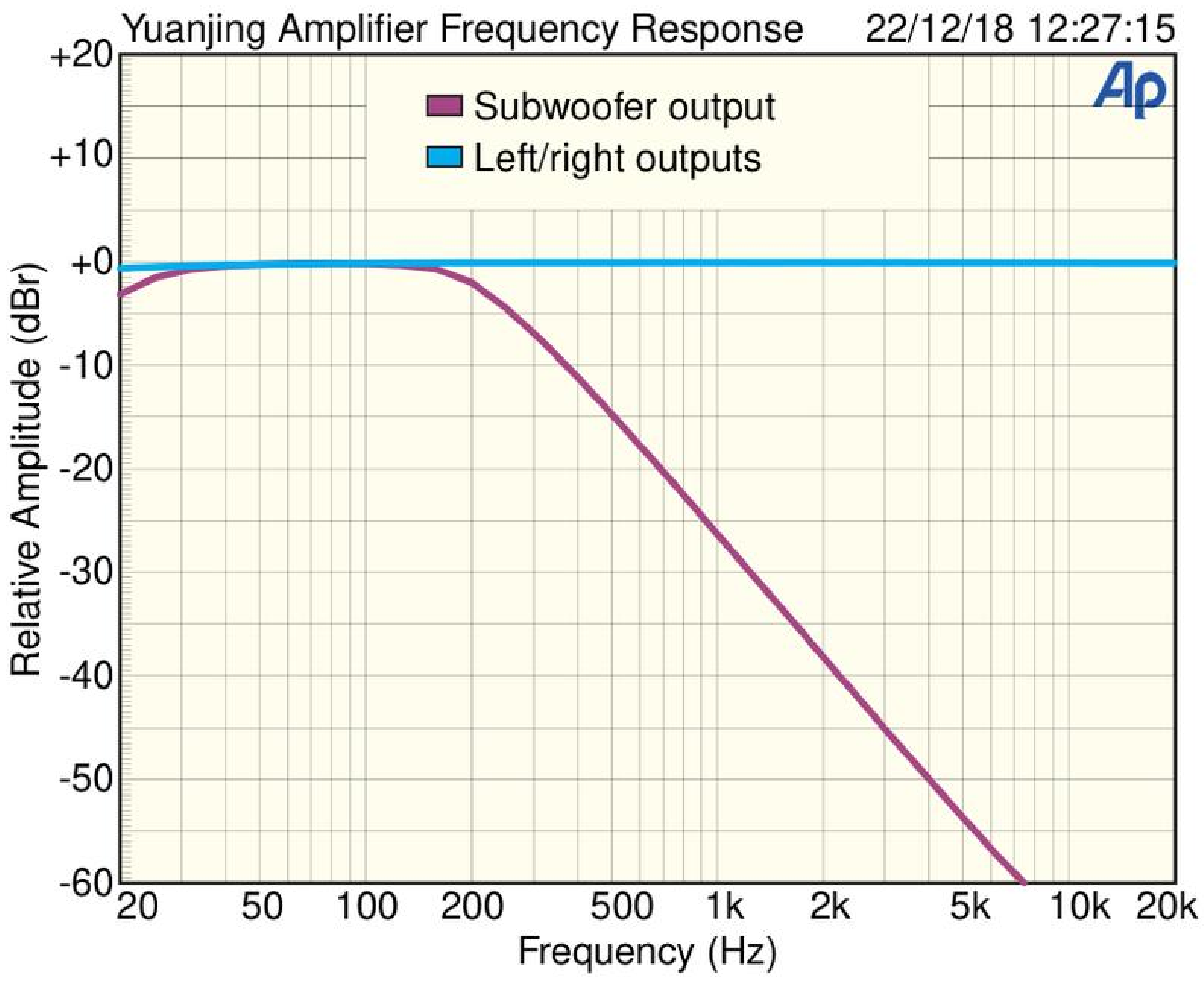


Fig.7: the frequency response of the Yuanjing-brand 4.1 channel amp is fine out-of-the-box, unlike the other one we tried. Note how its subwoofer low-pass filter is far less aggressive than the other board's, with significant amounts of low bass making it through, up to a few hundred hertz.

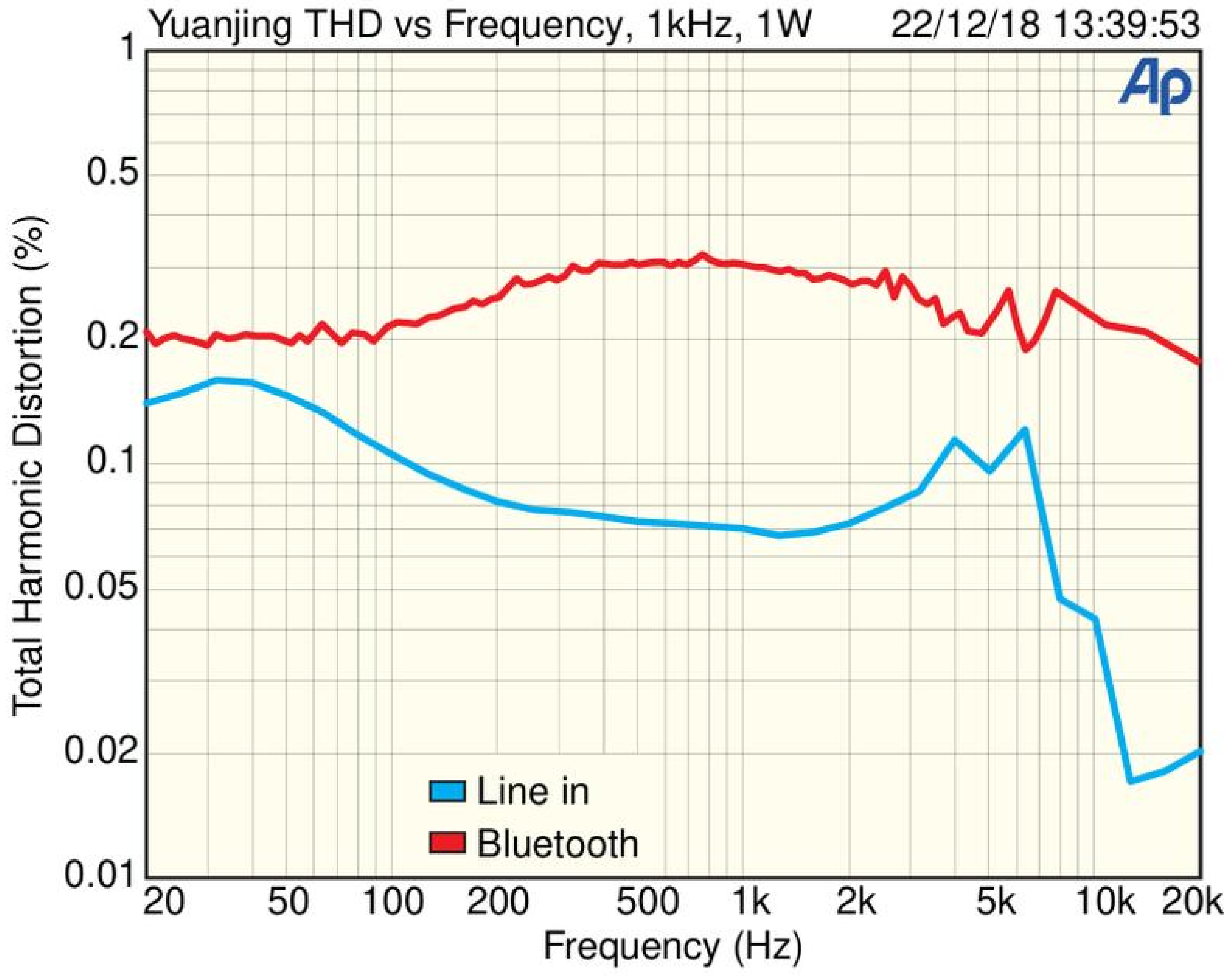


Fig.8: distortion performance is similar to the cheaper one; slightly worse at lower frequencies (probably due to the use of less-linear coupling capacitors), and slightly better at higher frequencies. Its performance is significantly better when using the line input pin header compared to Bluetooth, likely due to digital artefacts and noise in the output of the Bluetooth module.

Figs.7-9 show how the performance of the Yuanjing module compares. It's certainly usable as-is and is comparable to, or better than the XD172700 module in most areas.

Just one point to note: while this module comes with the appropriate pot nuts and washers (as seen in the photo) it neither includes the stand-offs nor the cute knobs which the other one has. Oh well – can't win 'em all!

Conclusion

These fully built and ready-to-go modules are very flexible and would have many useful applications such as in cars, TV soundbars, computer sound systems and amplifiers for smartphones. They should be very reliable due to their comprehensive protection against short-circuits and importantly, against overheating.

The fact that they only require a single DC supply and can run from 5V to nearly 30V makes them even more flexible. You can even get a few watts of audio output using a small USB charger.

The distortion, frequency response and crosstalk could all be improved, but for the price, we didn't expect super Hi-Fi performance.

These modules can easily be mounted inside a cheap Jiffy box or metal amplifier chassis. It's so straightforward, we aren't even bothering to give any instructions. Just mount them in the chassis, wire them up and away you go.

Reproduced by arrangement with *SILICON CHIP* magazine 2020.
www.siliconchip.com.au

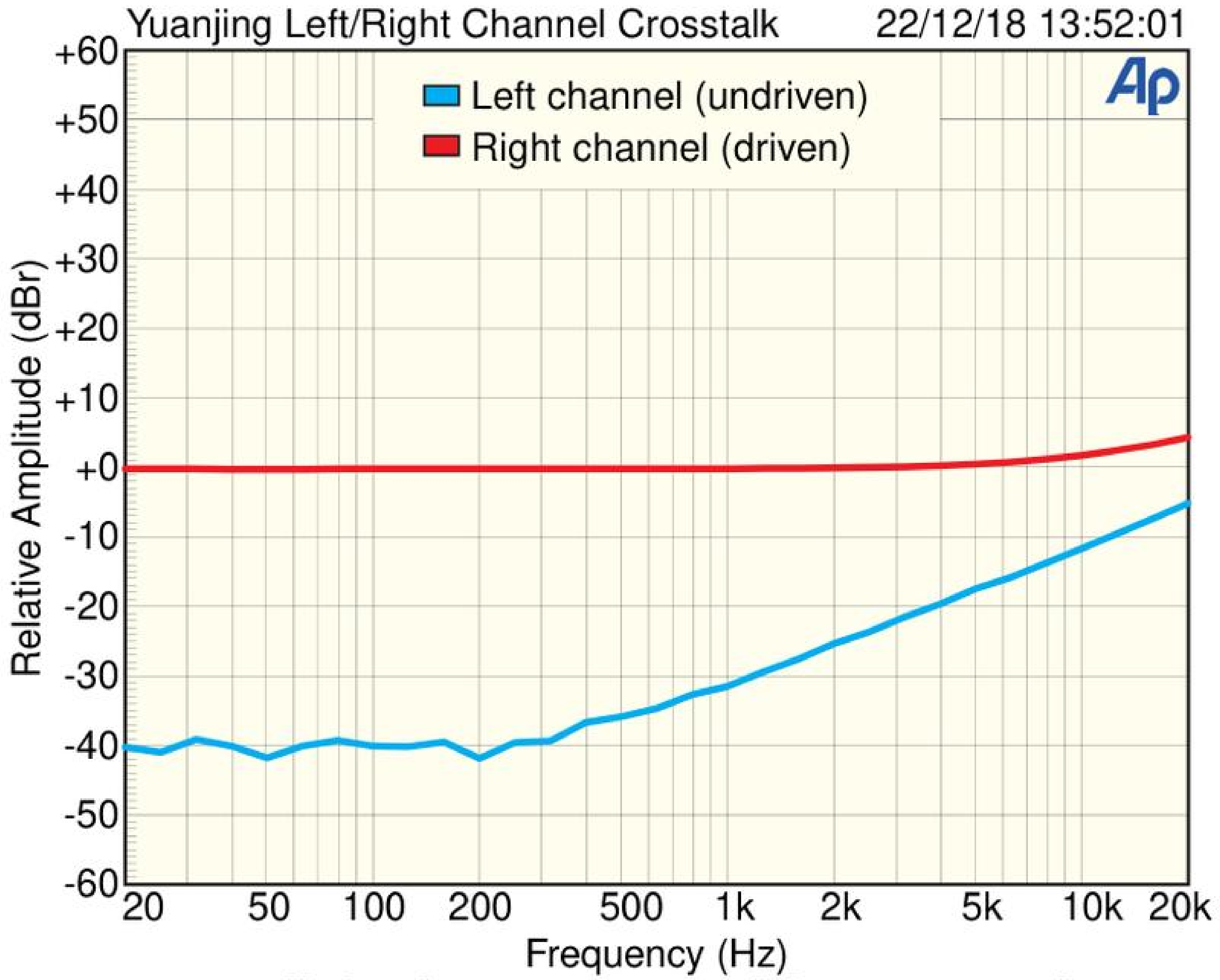
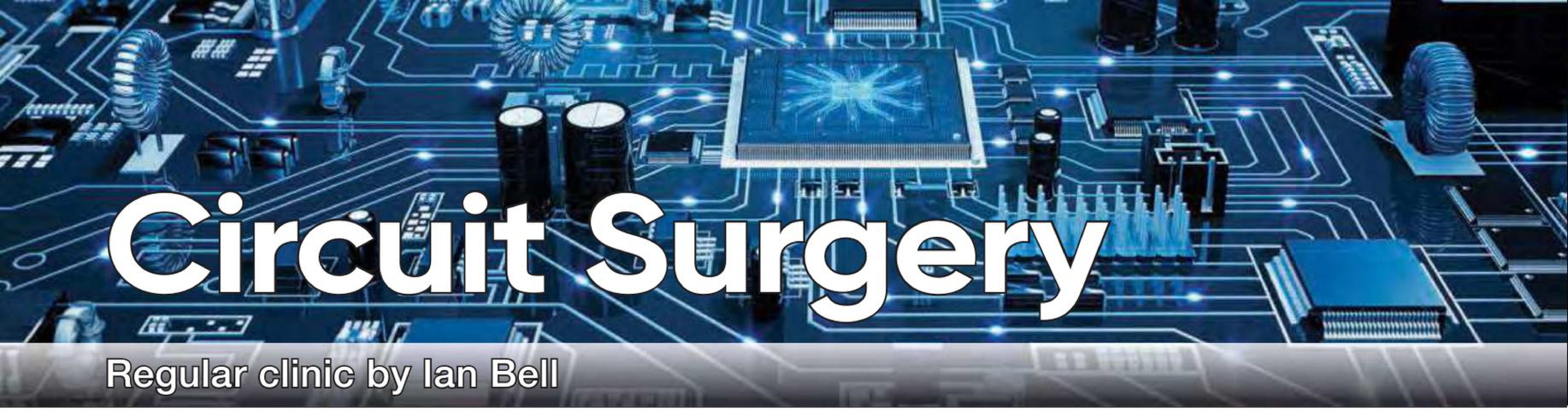


Fig.9: crosstalk for the Yuanjing amplifier isn't exactly great but it's significantly better than the cheaper one. You're not likely to notice this coupling when listening to ordinary program material with stereo speakers.



Circuit Surgery

Regular clinic by Ian Bell

LTspice – Using behavioural sources

Last month we started looking at LTspice sources, having previously used behavioural sources to draw waveforms to illustrate the article on class D, G and H amplifiers, and because I had been asked about getting waveform data in and out of LTspice for use with other applications. We looked at the basics of sources and the details of data input/export, including the use of WAV files. WAV files are a useful format for recording and importing for a wide range of circuits in LTspice – not just audio – but for audio circuits they have the advantage of enabling us to listen to the simulated waveforms. Following on from last month's introduction, this month we will look at behavioural sources in more detail. To make this more fun we will make use of the WAV export capability, illustrating behavioural sources by getting LTspice to synthesise a musical scale.

Recap – behavioural sources WAV files

As discussed last month, behavioural sources (LTspice BV and BI elements) facilitate the use of a large range of mathematical expressions to define their output. These expressions can involve time as well as the circuit voltages on any wire (to ground), voltage differences (between two wires) and the current in any element. Last month, we just used simple expressions, for example, the following source value equation will output a voltage which is 5000 times the current in resistor `R1`.

```
V=-5000*I(R1)
```

The LTspice behavioural sources provide basic operators such as addition and multiplication, logic operations (eg, `AND` and `OR`), conditional operations such as less than, and over 50 mathematical functions, including trigonometric functions, min/max, delays and random numbers.

The waveforms from WAV files can be input to an LTspice simulation by placing a voltage or current source on the schematic and setting the value to the source, eg:

```
wavefile=filename chan=channel
```

Here, *`channel`* is the channel number of the waveform in the WAV file. The filename can include the full path if the WAV file is not to be in the same folder as the LTspice schematic. To export waveforms, place a `.wave` directive on the schematic (using the the `.op` toolbar button):

```
.wave filename nbits samplerate
V(net) ...
```

Here, *`filename`* is the name of the WAV to be written to, *`nbits`* is he number of bits used for the wave data values (from 1 to 32), and *`samplerate`* is the sample rate of the waveform in Hz of the file from 1 to 4,294,967,295. The settings are followed by the list of net voltages to include in the file (the number listed determines the number of channels). You can think of this directive creating a virtual analogue-to-digital converter which writes to the file.

In last month's article we also discussed the requirements for using WAV files correctly. The amplitude of the waveforms from WAV files is limited to ±1V, which means it is often necessary to scale the waveforms to match them to/from the circuit's signal levels. This can be done using behavioural sources, as demonstrated last month. Inputs are straightforward as the WAV ±1V range is known, but outputs from it may be necessary to run the simulation twice – the first time to measure the signal's peak and the second with the scaling correctly set.

It is useful to be able to view and manipulate the WAV files outside of LTspice; for example, to check that the waveforms in the WAV are as expected and correctly match the simulation. A useful tool for this is Audacity, a free audio editor and recorder. Audacity can display waveforms and play the audio content. It also has numerous processing capabilities, including signal normalisation, which is useful in preparing inputs to simulations to cover the full WAV signal range.

Using Behavioural Sources

The most straightforward use of behaviour sources is in creating complex waveforms as inputs to your circuits, where the basic sources are unable to generate them. Another common use is as part of accurate models of circuits like op amps (macromodels) that do not reveal all the details of the component level design – this is often done by device manufactures. A third use is in the process of creating a new design – to model the ideas at an abstract level, rather than designing a full schematic. Such behavioural design is common in professional engineering and in some cases, such digital circuits are described in languages such as VHDL and Verilog, the detailed design can be created from the behavioural design automatically by software.

ADSR idea

As an example of a behavioural design, but mainly just to illustrate some of the capabilities of these behavioural sources, we will describe the creation of an LTspice simulation that represents the behaviour of a sound synthesis circuit similar to that which you might find in an analogue musical synthesiser. Regular readers will recall the *MIDI Ultimate Synthesiser* project (*PE*, February to July 2019). This synthesiser is typical in using an ADSR (Attack Decay Sustain Release) circuit to shape the envelope of the sounds it creates. The envelope (see Fig.1) is the variation of the amplitude of a single musical note, or percussive sound, with time. The amplitude initially rises to a peak (attack) and then deceases (decay) to a level which remains constant for a while (sustain) until the amplitude decreases to zero when the sound completes (release). Suitable choice of speed of attack, decay and release, and the level of sustain allow

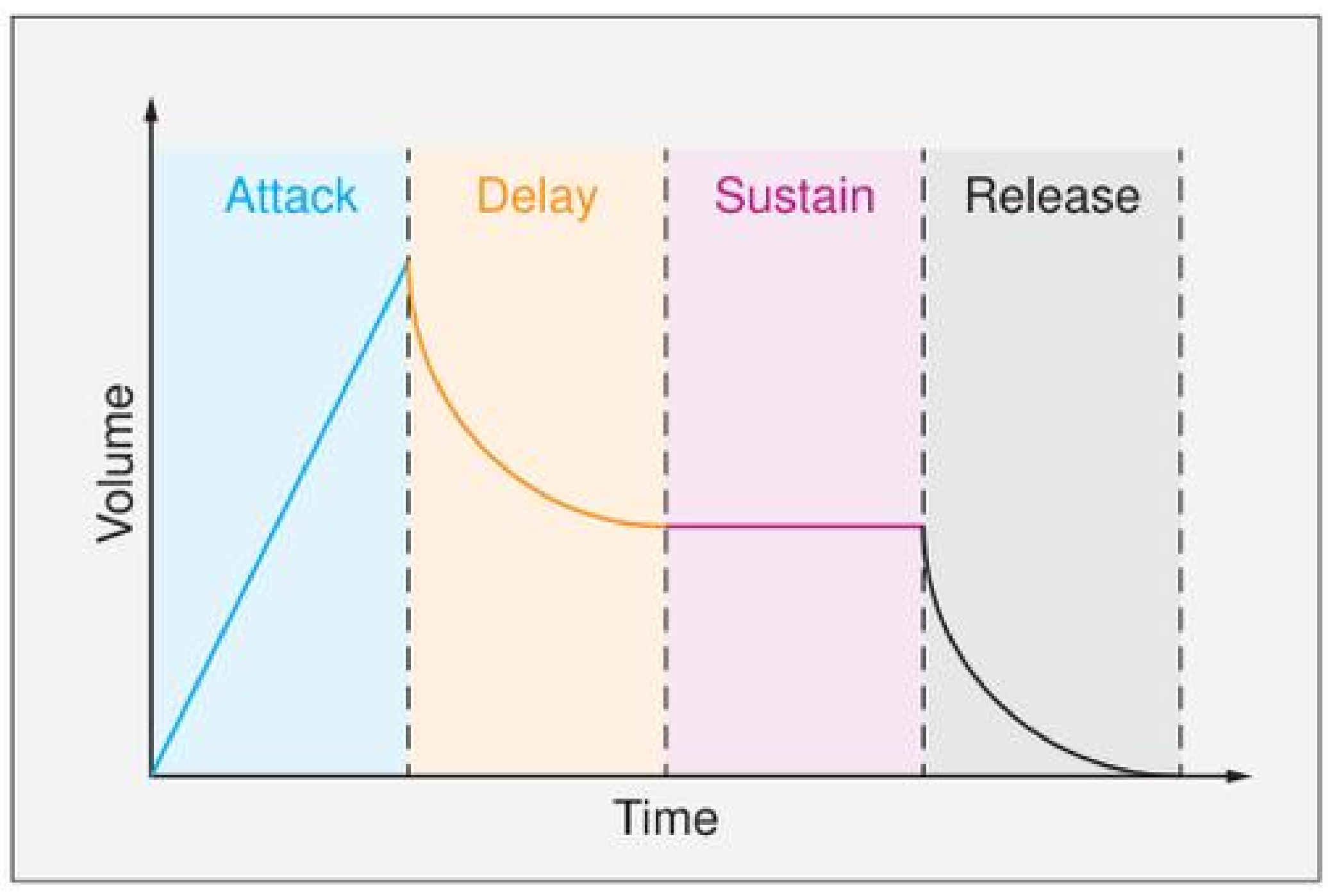


Fig.1. ADSR amplitude envelope.

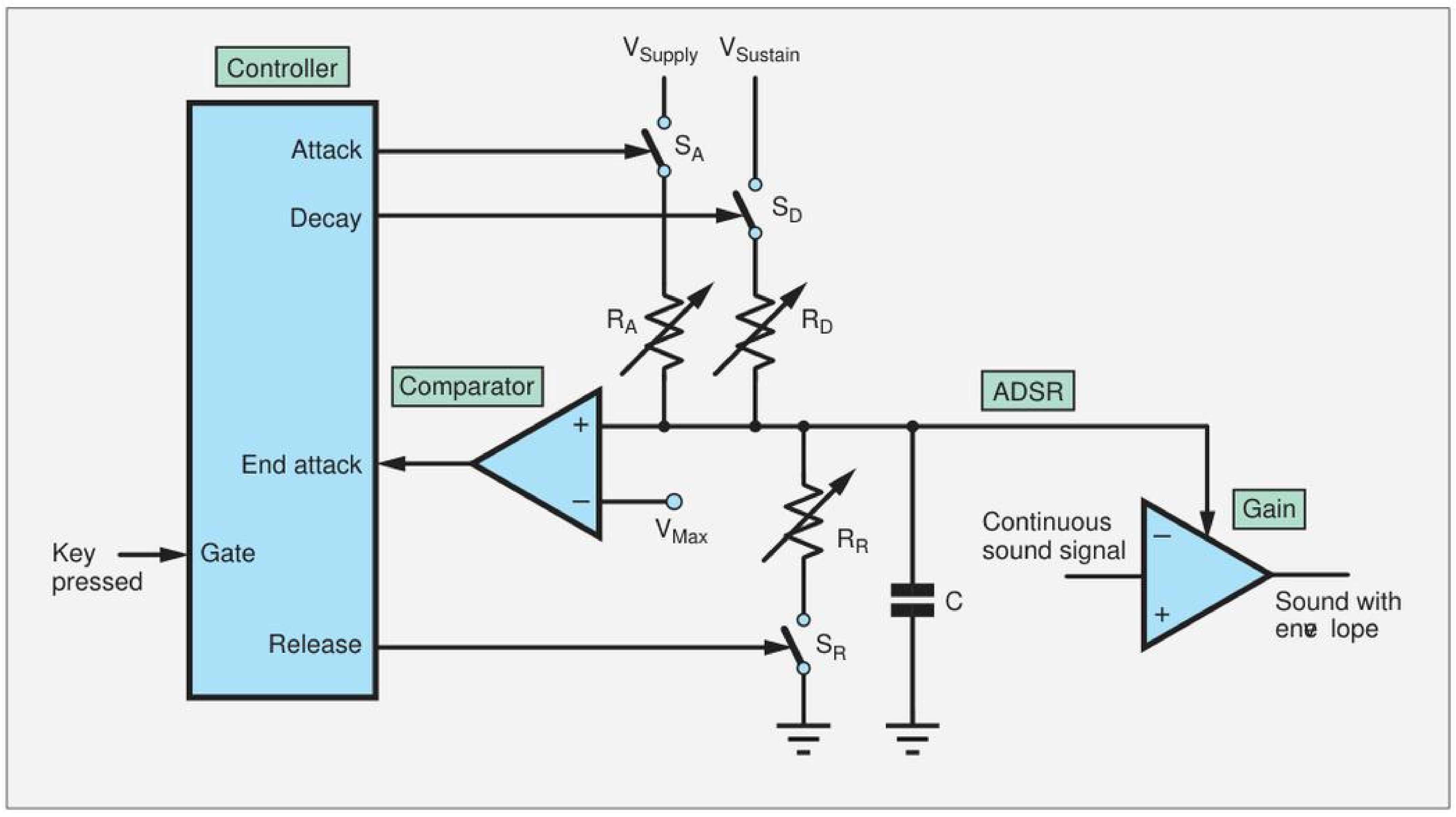
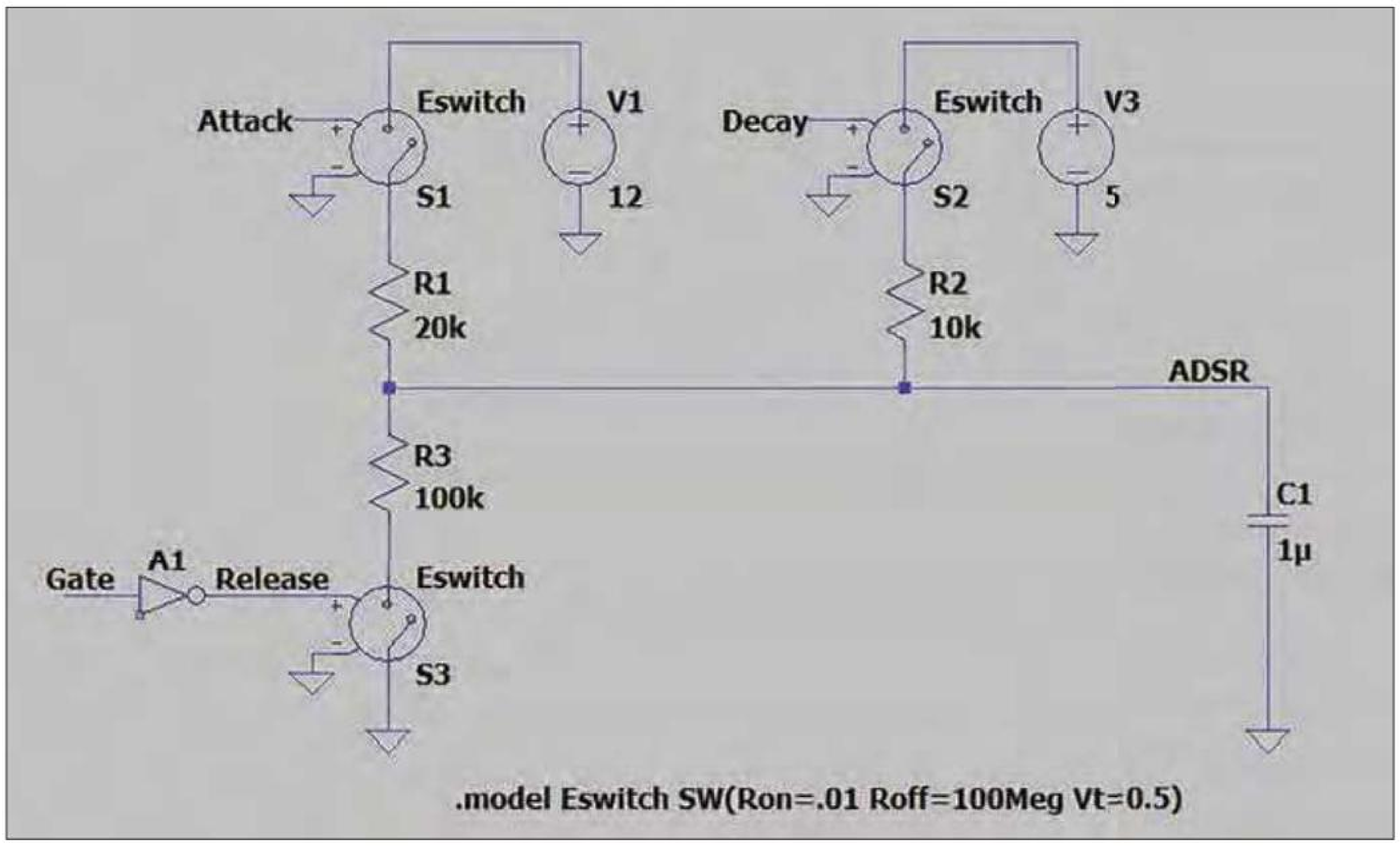


Fig.2. Music synthesiser ADSR envelope concept circuit.



```
.model Eswitch SW(Ron=.01 Roff=100Meg Vt=0.5)
```

Fig.3. The resistor, switch and capacitor network for the ADSR circuit.

the desired quality of sound to be achieved; for example, to help mimic a physical musical instrument.

The ADSR circuit in the *MIDI Ultimate Synthesiser* uses over 60 components and also requires further circuits to generate the sound signal, apply the envelope to the signal and create the trigger signal that indicates when a note is played. Imagine that the idea of an ADSR circuit was new and you wanted to develop it from scratch – there would be no existing circuits to borrow from. You could try to build a complete prototype, maybe with a few hundred components. If it did not work well – maybe there were problems with the basic design concepts, or the detailed implementation – then debug may be very difficult. However, if you do not mind working with some mathematics then the design could be developed first in behavioural form to evaluate and hone the basic ideas before creating the full circuit.

ADSR concept circuit

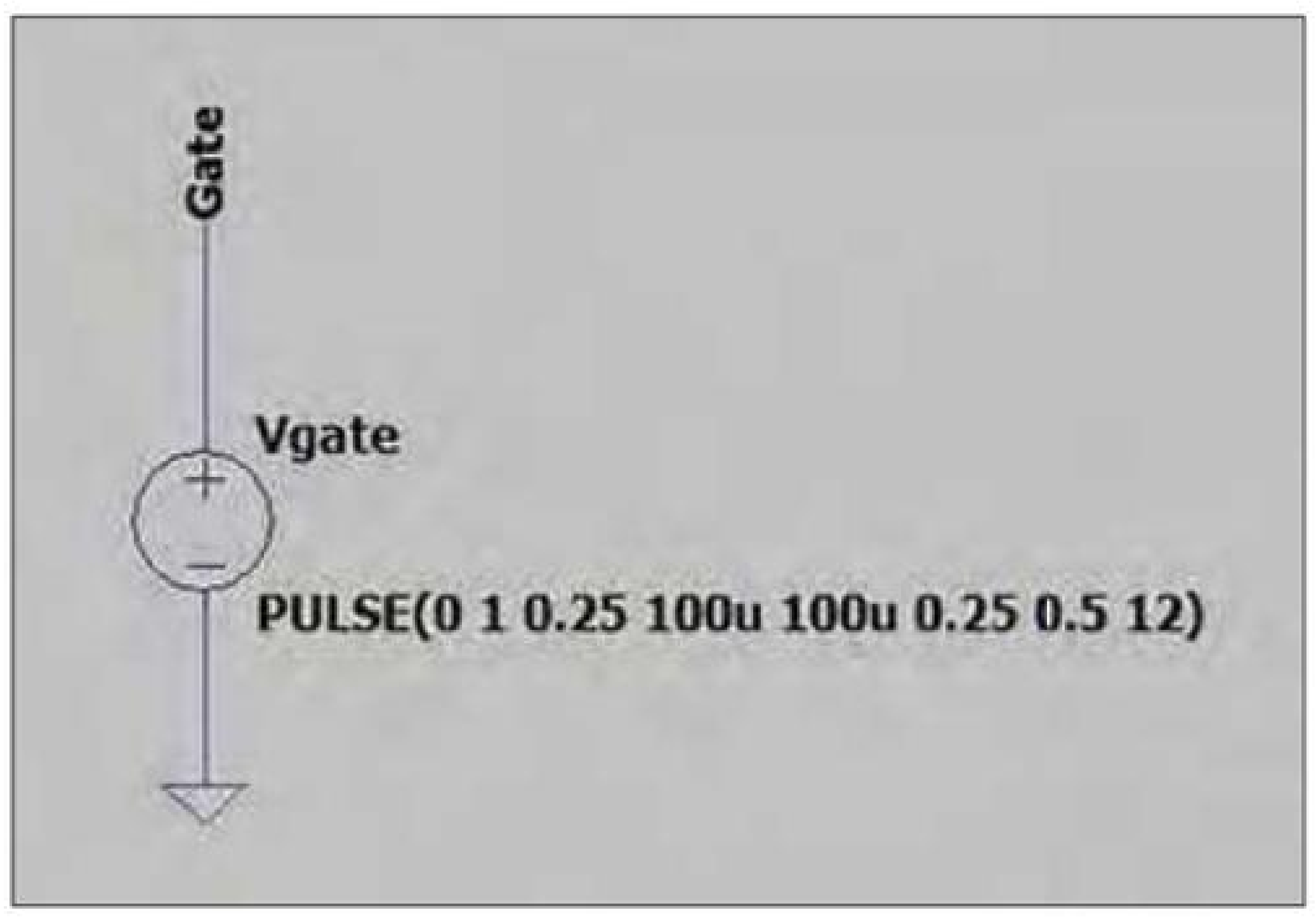
A concept schematic of the ADSR circuit is shown in Fig.3. We will use a mixture of real 'components' (resistors, capacitors and switches) and behavioural sources to implement the circuit – the choice is based on finding the easiest way to model the behaviour. The sound signal source, control circuit, comparator and voltage-controlled amplifier (VCA) will be largely implemented with behavioural sources.

Our ADSR circuit model follows the same basic principle as the one in the *MIDI Ultimate Synthesiser*, although slightly simplified. It operates as follows. The timing of the attack decay and release phases of the envelope are controlled by the charging or discharging of the capacitor (C) via the variable resistors R_A, R_D and R_R respectively (see Fig.3). The voltage on C (labelled 'ADSR' in the figure) is used to control the envelope of the sound via the VCA. When no note is being played the key pressed signal will be off (logic 0), which will cause the controller to switch on its release output (and turn the attack and decay outputs off). This will turn on switch S_R, discharging C through R_R. After the release time, the ADSR voltage will be close to zero and no signal output will occur. When the next note is played the key-pressed signal goes high, causing the controller to switch the release output off and the attack output on, switching on S_A and charging C through R_A towards the supply voltage. When the ADSR voltage reaches its maximum value (set by V_{max}) the comparator will switch, causing the controller to switch the attack output off and the decay output on. C will then discharge towards $V_{sustain}$ via S_D and R_D. If the key is pressed for sufficiently long, the ADSR voltage will level off towards $V_{sustain}$. This brings us back to the point where the key is released and C is discharged from whatever voltage it is currently holding via S_R and R_R. The shape of the envelope is controlled by the three resistors, the sustain voltage and length of key-press. In the original design C can also be switched to extend the range of timing.

Creating the simulation

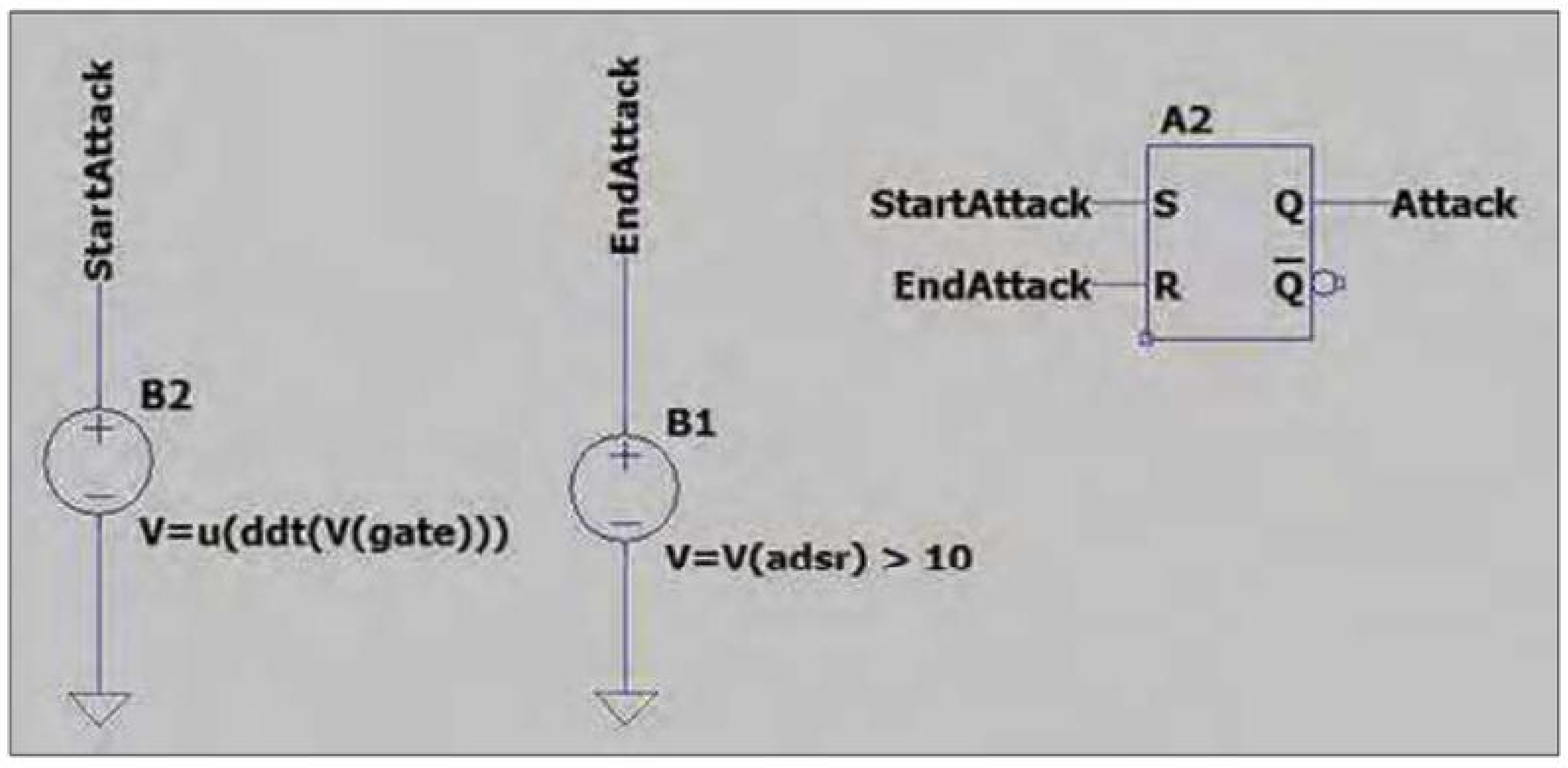
To create this simulation, we have to decide what 'music' the simulated synthesiser is going to 'play' – we will get it to create a twelve-note chromatic scale starting from the commonly used 440Hz reference note (A above middle C). The notes will all last a quarter second and be played every half second.

We start by drawing the resistor, capacitor and switch network using basic components (see Fig.3). This is derived directly from the relevant parts of Fig.2. The



```
PULSE(0 1 0.25 100u 100u 0.25 0.5 12)
```

Fig.4. Gate signal which produces twelve gate (key-pressed) pulses.



```
V=u(ddt(V(gate)))
V=V(adsr) > 10
```

Fig.5. The attack signal.

switches use SPICE S elements, for which we have to provide a model using a `.model` statement:

```
.model Eswitch SW(Ron=.01 Roff=100Meg Vt=0.5)
```

This defines a model called `Eswitch` (electronic switch) that is close to ideal in that it has an on resistance of 0.01Ω and an off resistance of 100MΩ. The switch is off when the control input voltage is below the threshold (`Vt`) of 0.5V, and on when it is above the threshold.

The release switch (S_R) is controlled directly by the gate (key pressed) signal. When the gate is off the release switch is on and vice versa – this is a logic NOT operation which is implemented using LTspice's behavioural logic elements. These may be another topic for another month, for now we just need to know they provide standard logic functions, do not need supplies and by default use a 1V logic signal – which is why the switches were configured with a 0.5V threshold (half the logic voltage).

The timing of the twelve regularly spaced notes can be created using a standard voltage source configured in pulse mode (see Fig.4). The pulses are 1V with a period of 0.5s and an on time of 0.25s. The first pulse goes high after a delay of 0.25s. The rise and fall times of transitions are 100µs.

Control of the attack and decay switches (S_A and S_D) is little more complex and is where we will start to see use of behavioural sources. We also need another logic element – a set-reset flip-flop (see Fig.5), which also features in the design of the original synthesiser. When the gate signal switches on, we set the flip-flop, which in turn activates the attack switch. When the ADSR voltage reaches the maximum value (V_{max} in Fig.2) the attack phase is stopped by resetting the flip-flop. This requires that we generate two signals: one to start the attack phase and another to end it.

Edge detection

To start the attack phase, we need to detect the positive edge (0 to 1 change) of the gate signal. There are different ways to do this in a real circuit (flip-flop or *RC* circuit) but with behavioural sources we do not have to worry about the implementation, just the function required. When a positive edge occurs, there is a positive rate of change of the signal voltage. Mathematically, 'rate of change' is found by differentiating a function and LTspice provides a time-derivative function (`ddt`) to calculate this. We set the value of a behavioural source to:

```
V=ddt(V(gate))
```

Now the source will output a voltage equal to the rate of change of the gate signal. It will be zero except when the pulse is changing – detecting the edges.

With setup of the `Vgate` pulse source described above, the edge change is 1V in 100µs, which is 10kV/s, so this source will output ±10kV pulses for the 100µs periods while the gate pulse is switching. This will not prevent the circuit working – the behavioural flip-flip is not real and will work fine with a 10kV input, but we want to keep the control logic to 1V for consistency. Also, we have negative pulses (for the 1 to 0 changes), which again will not affect the flip-flop, but which we do not need. There is an LTspice function called 'unit step' (`u`) which is defined as outputting 1V if the input is greater than 0, otherwise it outputs zero. If we apply the derivative of the gate voltage to this, we will get 1V when the positive edge is occurring and zero at all other times. So, the source function becomes:

`V=u(ddt(V(gate)))` (See Fig.5.)

The signal to end the attack phase is simpler. For the circuit in Fig.2 we see this occurs when the ADSR voltage is greater than V_{max} (detected by the comparator). In the original circuit the supply was 12V (as it is in Fig.3) and V_{max} was 10V. We can create a signal that is 1V when the ADSR voltage is above 10V (0 otherwise) using a 'greater than' conditional operator (>)

```
V=V(adsr) > 10
```

As shown on Fig.5. You can use four conditional operators (> < >= >=).

Delays

Control of the decay switch *seems* straightforward. The decay switch is on when the gate signal is on (a key is pressed) AND we are not in the attack phase. We can detect the two conditions with `V(attack) < 0.5` and `V(gate) > 0.5`. LTspice has logical operators (AND: `&`, OR: `|`, XOR: `^` and NOT: `!`) so we can write the full condition for the delay signal as:

```
V=V(attack) < 0.5 & V(gate) > 0.5
```

Unfortunately, this will cause a simulation failure. The problem is due to the attack signal being controlled by the gate signal,

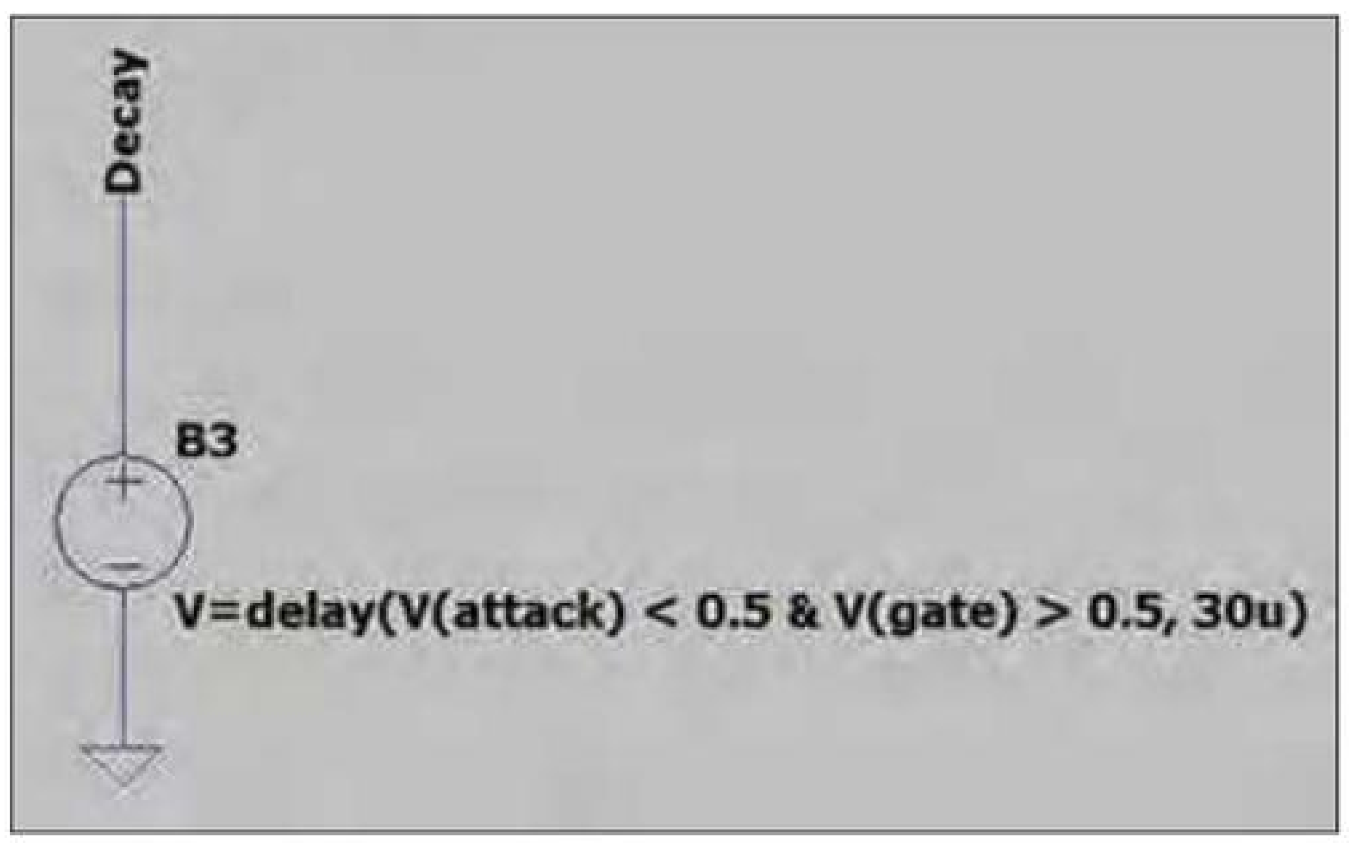


Fig.6. The decay signal.

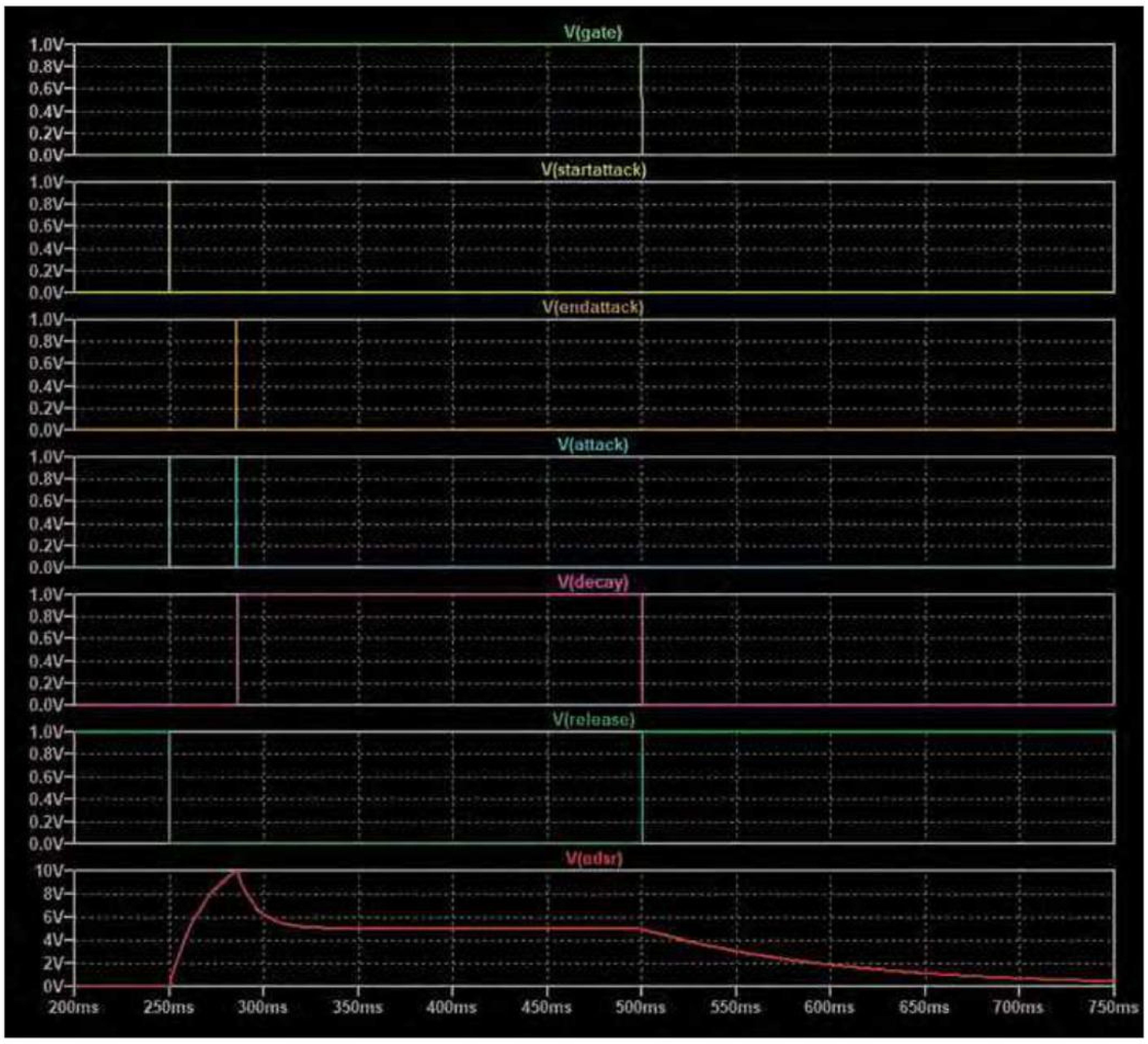


Fig.7. Simulation results showing the control signals and resulting ADSR envelope.

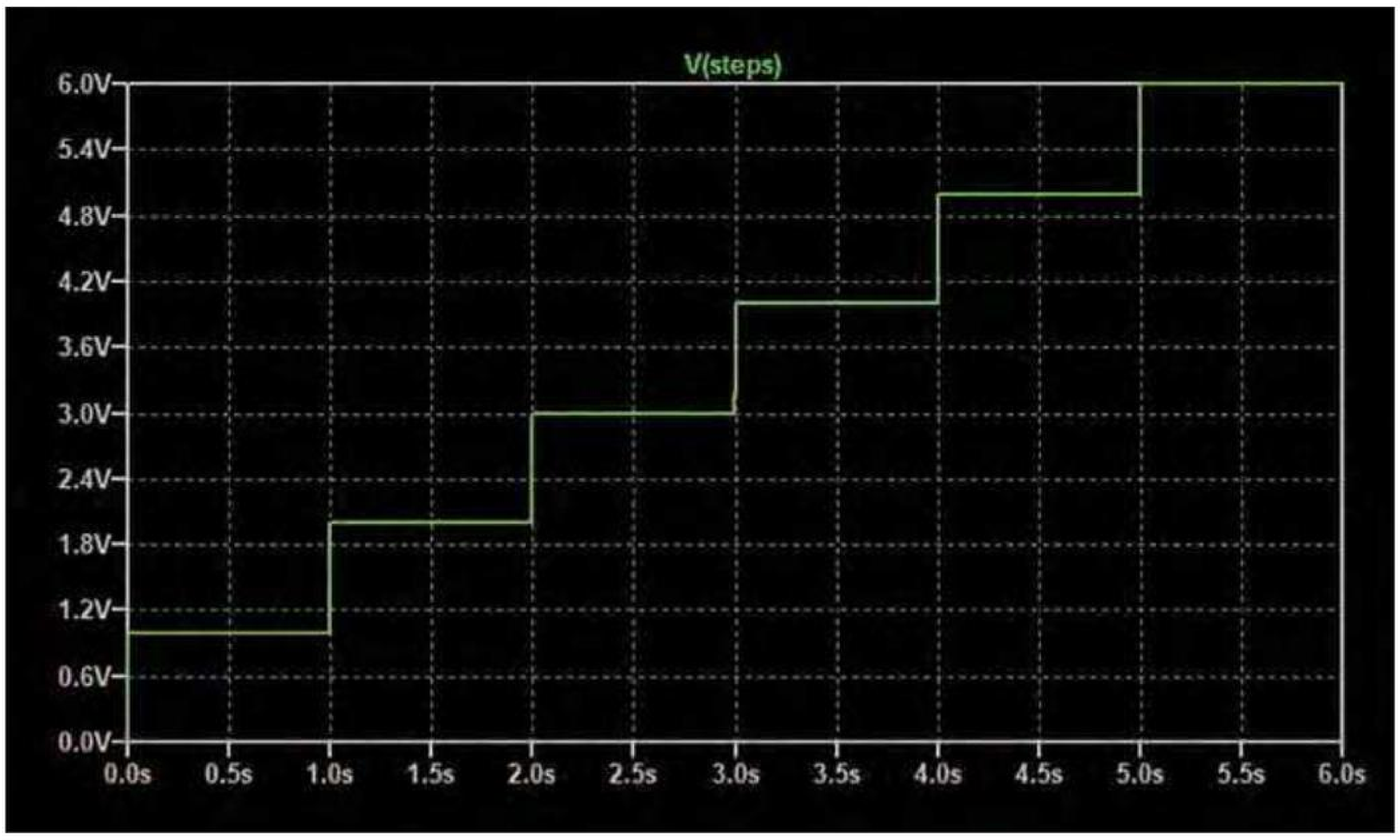


Fig.8. Stepped waveform using `V=ceil(time)`.

which is in turn controlled by attack. With these functions having zero delay (unlike anything in a real circuit) the simulation can lock up. The solution is to introduce a delay – LTspice has a function delay to achieve this. If we have a voltage `V(sig)` we can create a version of this signal delayed by time *`tdelay`* using:

```
V=delay(V(sig), tdelay)
```

Using delay tends to slow down the simulation, particularly if the delay time is short compared with other activity. For this circuit 300µs worked. See Fig.6, where the source expression is:

```
V=delay(V(attack) < 0.5 & V(gate) > 0.5, 300u)
```

Running a simulation with all the elements from Fig.3 to Fig.6 produces the results in Fig.7, which shows a single ADSR cycle. We can play with the values of `R1`, `R2`, `V3` and `R3` (Fig.3) and the on time for `Vgate` pulse source (Fig.4) to change the envelope shape.

We now have an envelope but no tone signal to apply it to. In the real synthesiser this may come from a voltage-controlled oscillator (VCO). We can do something similar with behavioural sources – that is, create a wave whose frequency is controlled by a voltage. We will start with a sinewave as this provides a single fundamental frequency (the pitch of a musical note). We could extend this to create waveforms with different sound qualities (timbre in musical terms) by adding harmonics (multiples of the fundamental frequency).

Sines and times

The sine function is familiar to many from school trigonometry. In that context we typically take an angle θ and find its sine, written as sin(θ), in which the angle is measured in degrees. If we plot a graph of sin(θ) against θ we get the familiar sinewave shape, which repeats every 360°. However, the degree is not the only unit for measuring angles and in mathematics, and LTspice's trigonometric functions, angles are measured in radians. The conversion is straightforward: 360° is 2π radians, so sin(θ) repeats every 2π with θ in radians (see **https://en.wikipedia.org/wiki/Radian** if you are new to radians).

For generating a sine waveform we need to apply a value to the sine function which varies with time. For a frequency of f Hz we need the sine function to repeat every $1/f$ seconds. If we use sin(t), where t is time the wave will repeat every 2π seconds. If we multiply time by 2π, that is use sin($2\pi t$) the wave will repeat every second (when $t = 1$ we evaluate sin(2π)). To set a different frequency we multiply $2\pi t$ by the frequency, that is use sin($2\pi ft$). For example, for 100Hz we have sin($2\pi \times 100 \times t$), we evaluate sin(2π) at $t = 1/100$s, so the waveform repeats every 1/100th of a second as required.

Translating sin($2\pi ft$) into LTspice syntax we can generate a 100Hz sinewave with a behavioural source using:

```
V= sin(2*pi*100*time)
```

Note that `pi` and `time` are keywords recognised in LTspice expressions, `time` is a value equal to the current simulation time. We can use a voltage to set the frequency simply by replacing the fixed frequency value in the above expression with a reference to that voltage, for example:

```
V= sin(2*pi*100*time*V(freqcontrol))
```

Scales

Given that we are aiming for a chromatic scale of musical notes we need to generate signals of the correct frequencies. The frequency of a music note, f_n, which is N steps away from a reference frequency f_0 can be found using the formula:

$$f_n = f_0 \times 2^{N/12}$$

We can write this formula in LTspice syntax. The multiply and to-the-power-of operators are `*` and `**` respectively, in common with many programming languages. With f_0 = 440Hz we could use:

```
V=440*2**(N/12)
```

This expression creates a voltage numerically equal to the frequency we want. To make this work we need a value of

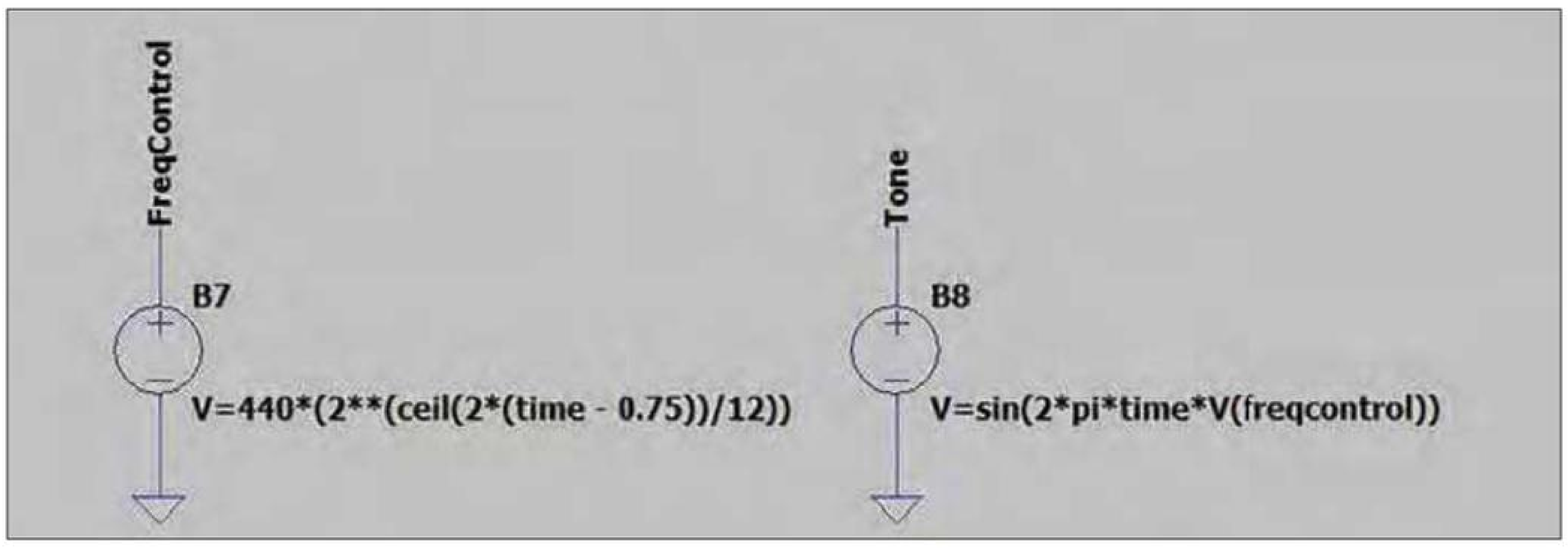

FreqControl
B7
V=440*(2**(ceil(2*(time - 0.75))/12))
Tone
B8
V=sin(2*pi*time*V(freqcontrol))

Fig.9. Creating the tones of the chromatic scale.

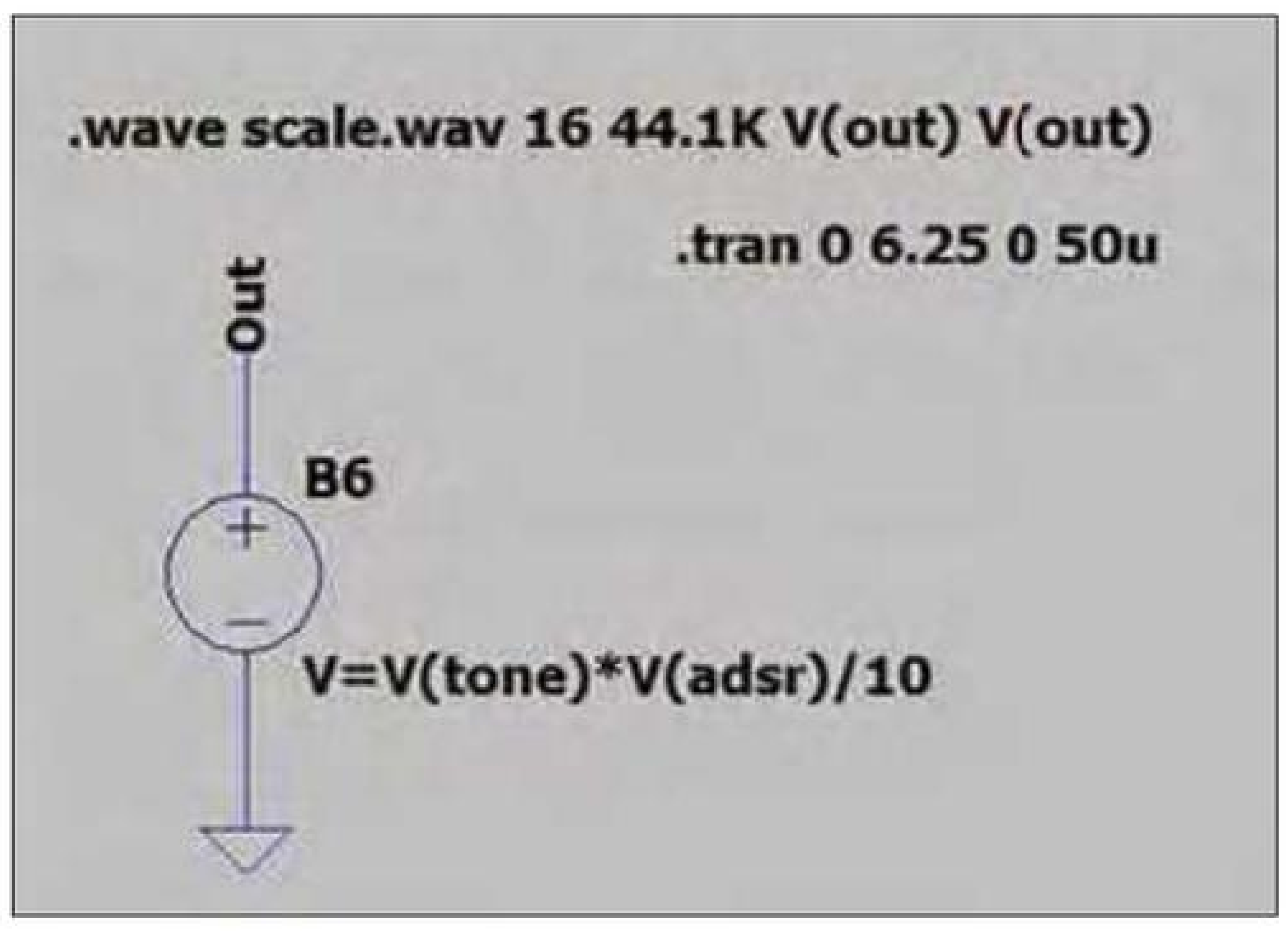


Fig.10. Final output source – this takes on the role of the VCA.

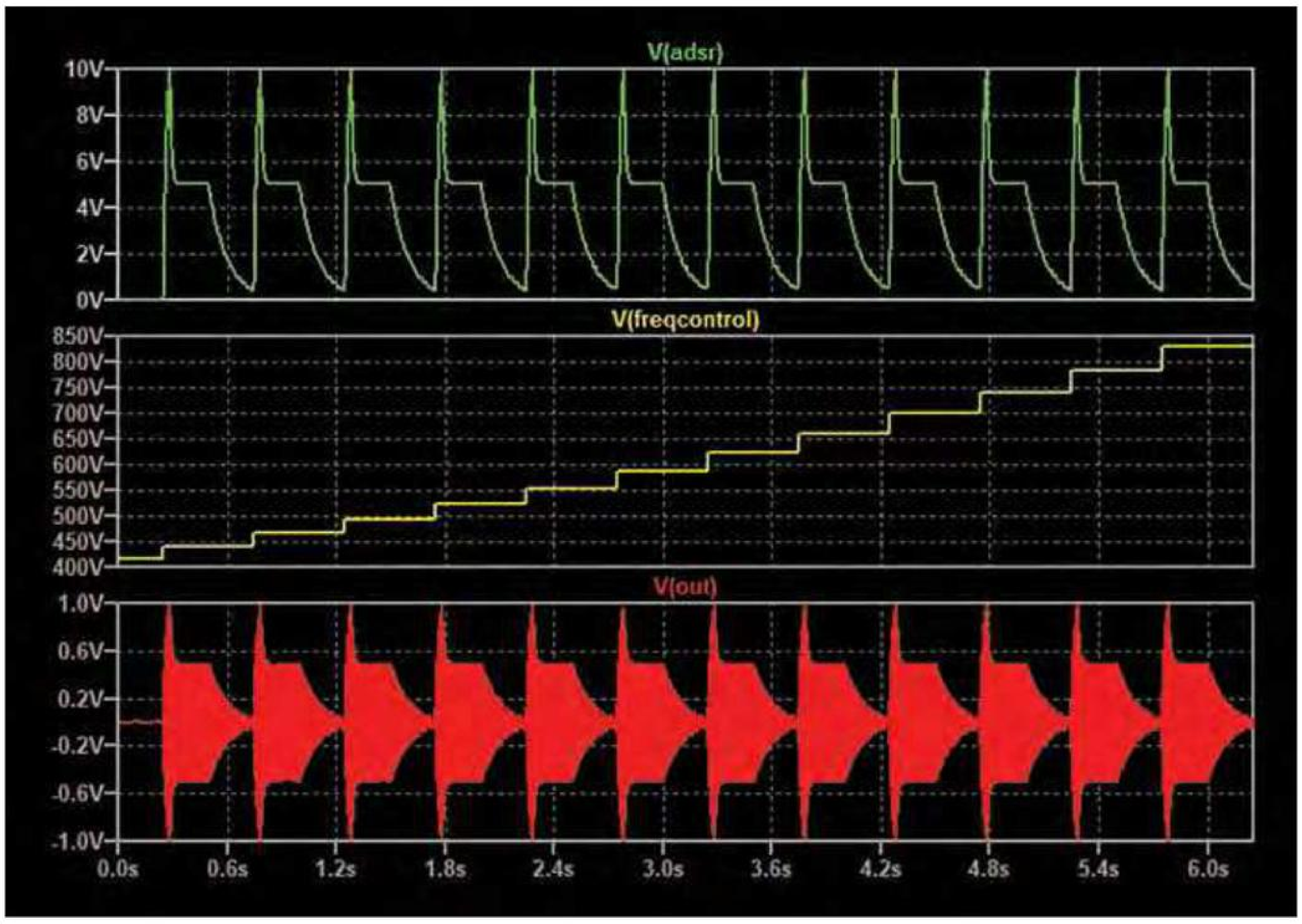


Fig.11. Complete chromatic scale of notes with ADSR envelopes.

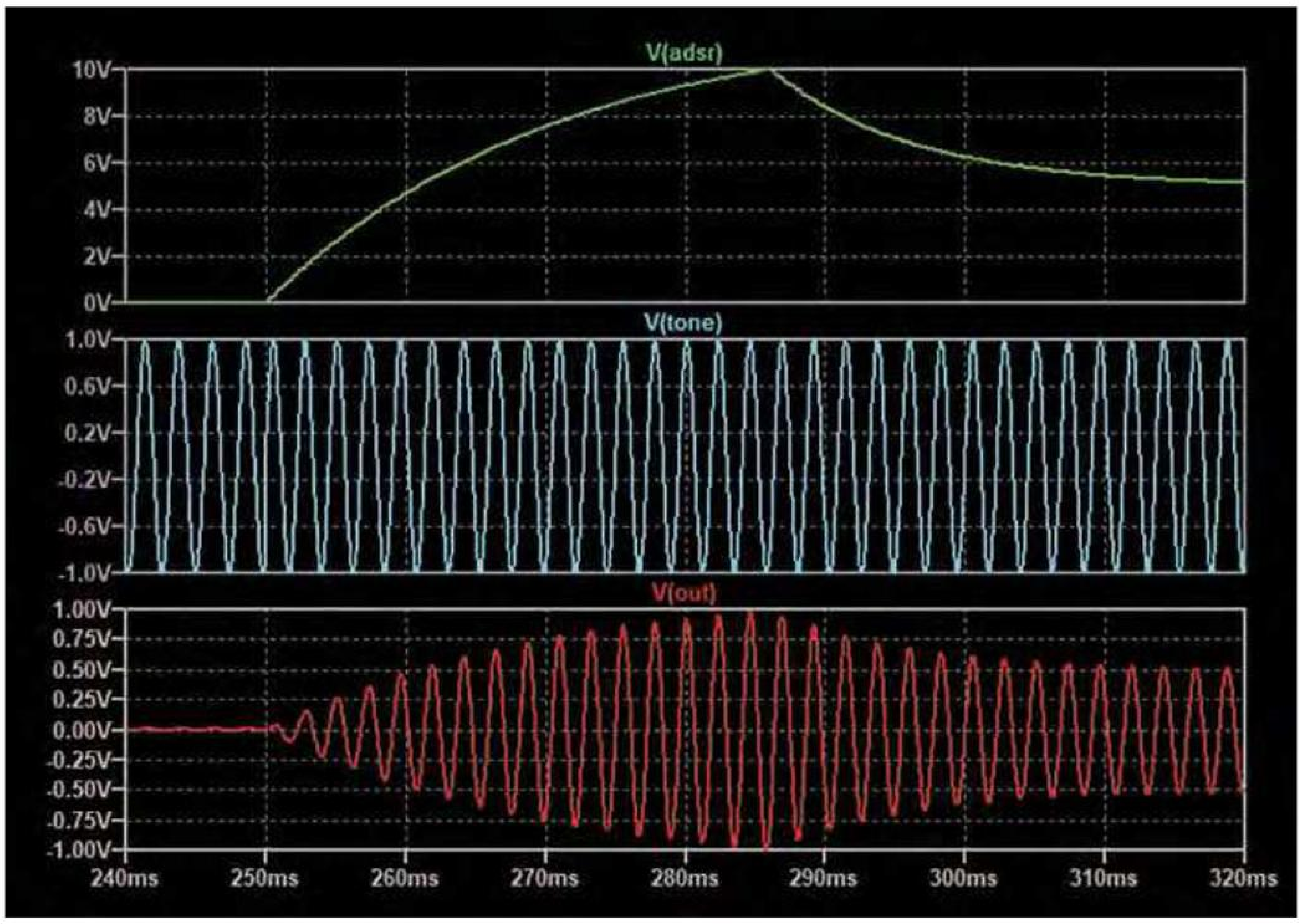


Fig.12. Zoom-in to show tone signal and part of one note envelope.

N, which must be an integer (whole number); for this we can use an integer voltage value (1V, 2V…). Given that we want to run up the musical scale we need a voltage which steps through integer voltages with time. We can create a voltage source which directly depends on time:

```
V=time
```

This will create a linear ramp increasing at 1V/s (volt per second), but not the steps we need. To create the steps, we can round the voltage of this ramp to the nearest integer using the LTspice `ceil(x)` function, which outputs an integer equal or greater than `x`. Using the following we get the waveform shown in Fig.8:

```
V=ceil(time)
```

For our chromatic scale (12 notes in six seconds) we need to step at twice this speed, which we can do using 2 × `time`. Also, to coordinate with the note changes, which start at 0.25s, we need to shift the waveform in `time`, which can be done by adding or subtracting a value from `time`. Specifically, we want the fundamental note ($N = 0$) to start at 0.25s. If we subtract 1.5 from 2 × `time` the value from the `ceil` function will start at –1 (at `time = 0`) and switch to 0 at 0.25s. So, for our *N* value we can use the following expression:

```
V=ceil(2*time-1.5)
```

We not have to use a separate voltage source for this. We can substitute this expression into the frequency-control voltage expression above to get:

```
V=440*(2**(ceil(2*time-1.5)/12))
```

The above discussion leads to us adding two voltage sources to the schematic – one to generate the frequency control voltage and the other to produce the musical tones. This is shown in Fig.9.

Envelope

We now have the tone signal and need to apply the envelope to it (the function of the VCA in the real system). This is easy to do by multiplying the tone by the ADSR signal. Given that the maximum amplitude of the ADSR signal is 10V and the maximum of the sine function is 1, this will result in a 10V peak tone-with-envelope signal. It would be fun to output the signal to a WAV file so we can listen to the effects of changing the ADSR parameters. As discussed last month, voltages written to the WAV file need to be limited to ±1V, so we need to divide the 10V signal by 10. From this the expression for the final output voltage source is:

```
V=V(tone)*V(adsr)/10
```

This is shown in Fig.10, which also includes the `.wave` and `.tran` directives.

Putting all parts from Fig.3 to Fig.6, and Fig.9 and Fig.10 on one schematic, and simulating produces the results shown in Fig.11. The tone signal details cannot be seen at this scale so, Fig.12 shows a zoom-in to part of one note. We can listen to the scale using an audio player and load it into Audacity to check the WAV file waveform is correct.

This simulation was contrived to illustrate use of behavioural sources, so the approach may not be the best way of doing things. For example, changing it to produce a different note sequences would be difficult – it may be better for the notes to be defined by a PWL input file. The simulation can be extended and improved in other ways too; for example, adding harmonics to the tone waveform for timbre and using LTspice's random number functions to create noise waveforms for percussive sounds. With these improvement in mind, a creative reader could produce a musical composition synthesised by LTspice!

Simulation files

Most, but not every month, LTSpice is used to support descriptions and analysis in *Circuit Surgery*.
The examples and files are available for download from the *PE* website.

AUDIO OUT

By Jake Rothman

Low-noise Theremin Power Supply – Part 1

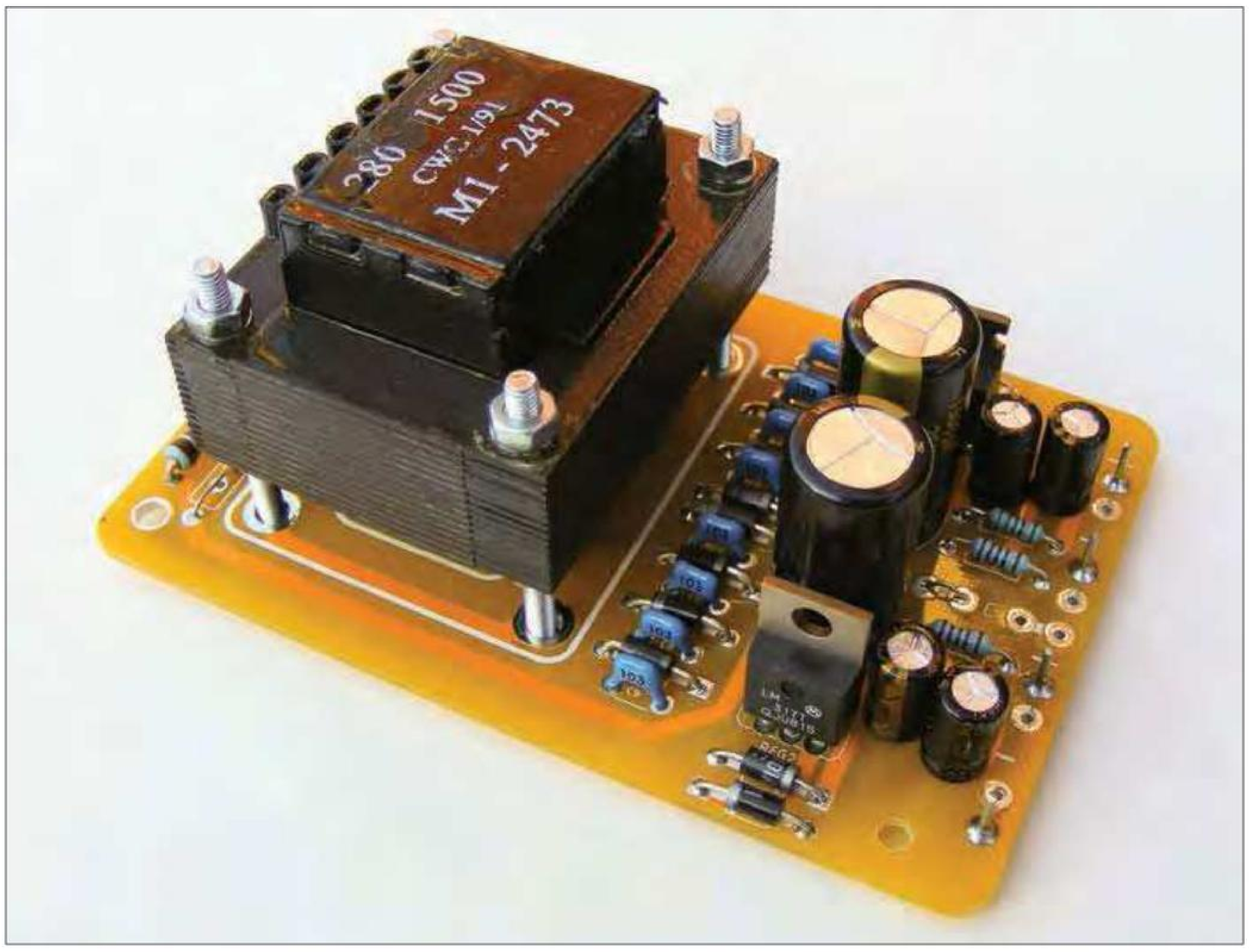


Fig.1. The completed Theremin power supply board, delivering +9V and +15V.

Oh no, not another power supply unit!, I hear you say. 'You can buy them online for a fiver and I've got 20 in the kitchen drawer'. True, if it's noisy, un-earthed black plastic 'wall-warts' destined for landfill that you want. Plug such a PSU into a Theremin, AM radio or Hi-Fi pre-amplifier and their performance will be degraded. Theremins generate a horrid 50Hz warble sound as the mains-related electromagnetic interference (EMI) emissions from cheap PSUs modulate the pitch. It's even worse if the PSU is a switch-mode design. Many SMPSs have 4.7nF Y capacitors connected from the mains input to 0V, guaranteed to make Theremins have an absolute fit. For most electronic equipment – such as PCs, TVs or printers – PSU RF (radio-frequency) emissions don't matter. The European regulations (CE, known cynically as 'Chinese Export') don't care about interference emitted below 1MHz. This is one reason there are so many disillusioned audio, and low-frequency radio engineers.

Low-noise design

There is one way out of this modulation misery and that is to build your own power supplies. In the quiet of the present lock-down, the uniquely low-noise of this design shown in Fig.1 will be even more noticeable.

The simplest way to reduce EMI-affecting circuits is distance, because fields obey the inverse-square law (double the distance, the intensity goes down four times (ie, two squared)). Although I dislike external PSUs, there is no substitute for distance to reduce noise. If you are building expensive audio/music gear, where an internal PSU is expected, then the use of a screened toroidal transformer is almost mandatory. For a Theremin housed in a big or long box, it's often possible to position the power supply out of the way.

Down to earth

To deliver maximum playing range, a Theremin should be earthed to complete

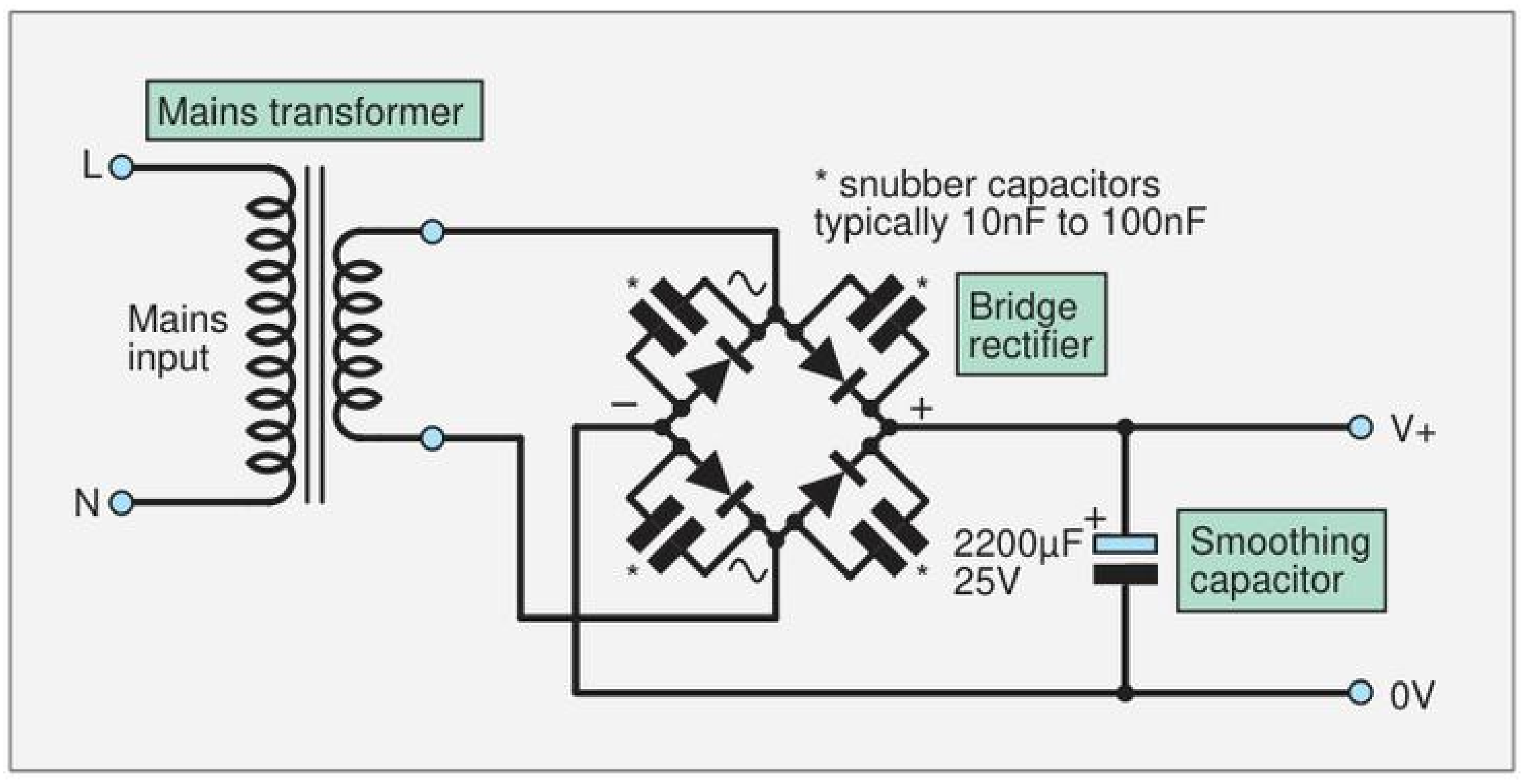


Fig.2. Snubber capacitors connected across the diodes in a bridge rectifier suppress switching spikes.

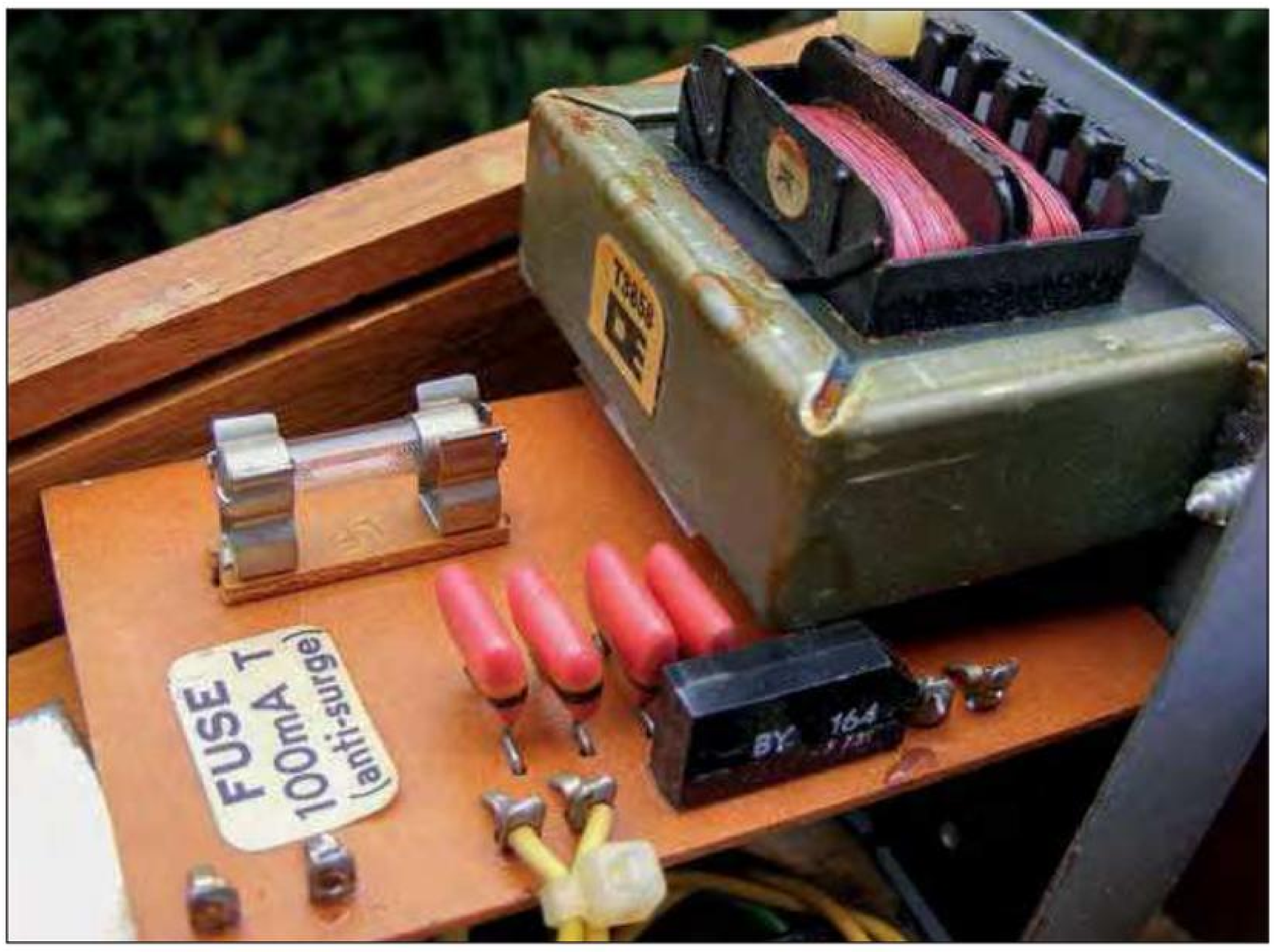


Fig.3. Example of snubber capacitors from an old Robert's radio – these are essential for hum-free reception on AM (long/medium wave).

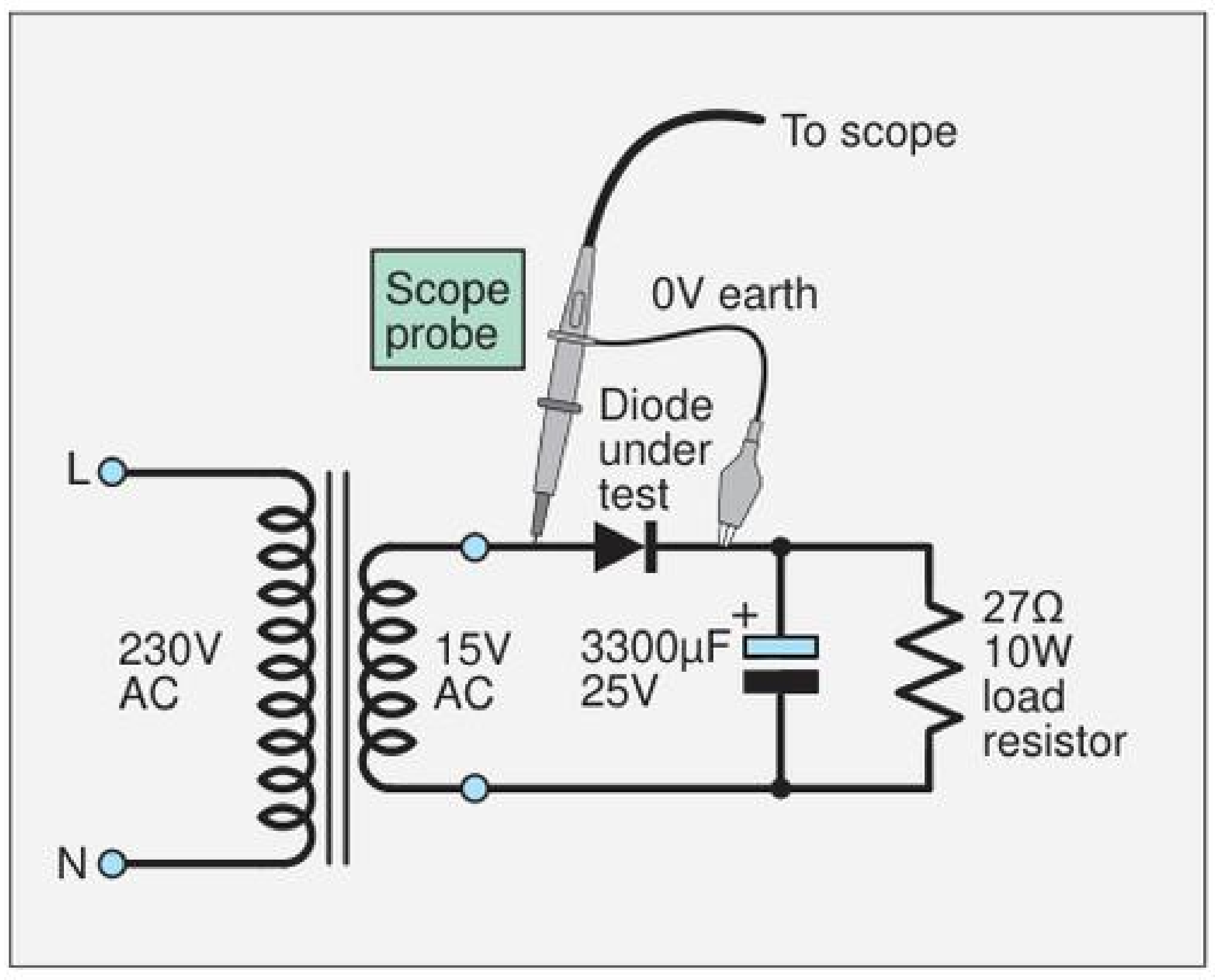

To scope
Scope probe
0V earth
Diode under test
L
N
230V AC
15V AC
3300µF 25V
+
27Ω 10W load resistor

Fig.4. Simple half-wave power supply used to compare rectifier diode switching spikes.

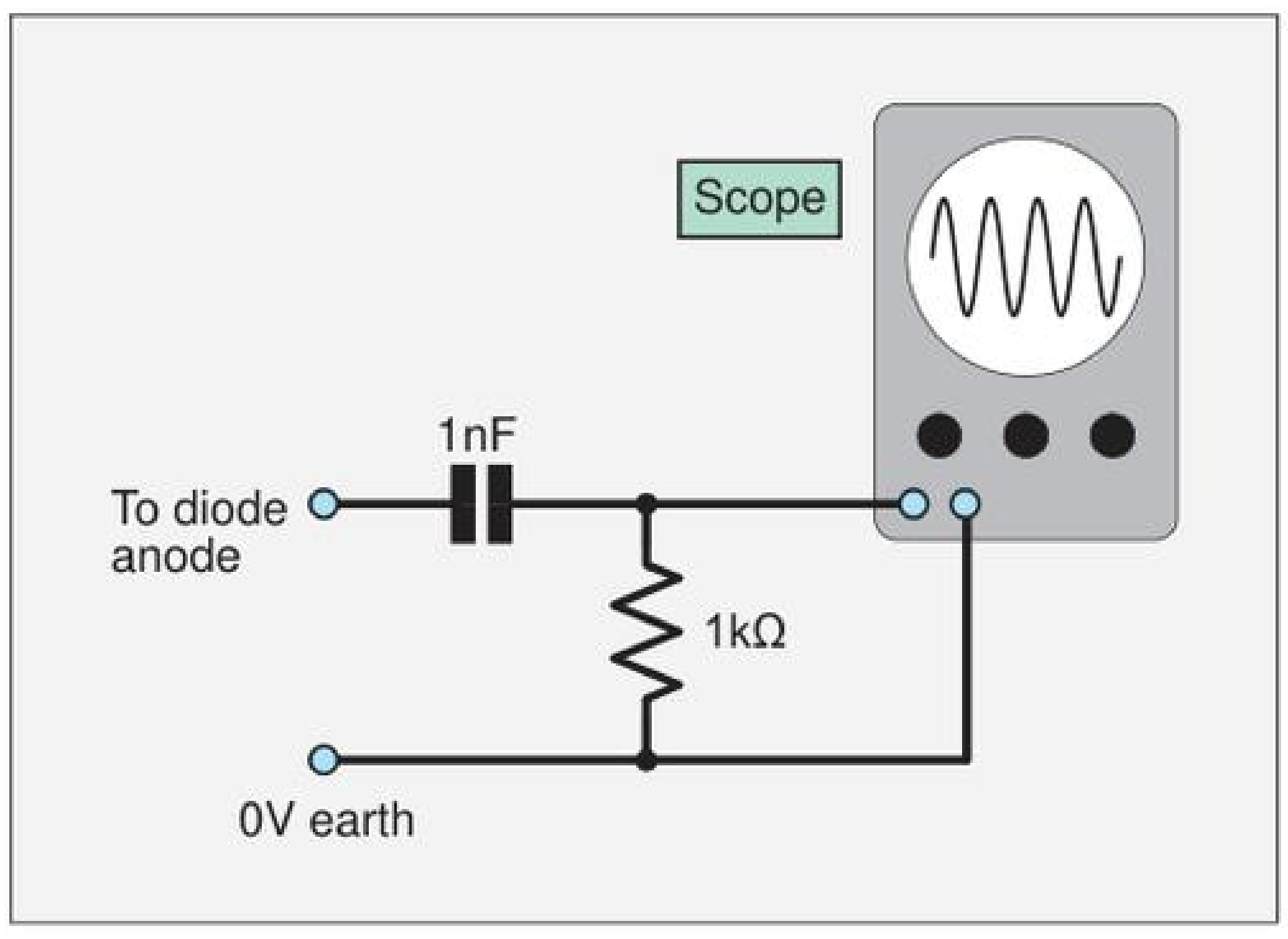

Scope
1nF
To diode anode
1kΩ
0V earth

Fig.5. High-pass filter on scope probe blocks 50Hz AC ('mains hum').

the capacitive circuit between the hand and the antenna. Most wall-warts have no earth at all; often having a plastic pin in the earth position on the plug, which easily breaks off. In the UK, this prevents it being plugged into the mains socket resulting in more e-waste. My power supply is – of course! – earthed via a proper three-pin socket and mains plug.

Rectifiers

In the old days of AM radios, the bridge rectifier diodes in the power supply always had snubber capacitors of around 10nF to 100nF connected across them (Fig.2) to stop what was known as 'modulation hum'. These are shown in the 1978 Robert's RM30 table radio supply in Fig.3. These were needed to suppress RF bursts produced by a sharp spike when the diode turned off. I've always added these capacitors as a matter of habit in all equipment. For this article, I decided to do a bit of investigation. These spikes are produced by the stored charge from the recombination of holes and electrons at the diode junction. This effect is especially pronounced with old slow silicon rectifiers, such as the good-old 1N4001.

This problem is especially bad with half-wave rectification, since the transformer has no load on half cycles where it is free to ring undamped. To examine the spikes, initially I used the half-wave rectifier circuit in Fig.4 since it only needs one rectifier. For experimentation, it is simpler to change one diode than the four of a full-wave rectifier. Also, with a full-wave rectifier the switch-off diode is damped the moment the other diode turns on, so the effect lasts a shorter time. It's a good idea when looking at this sort of thing to put a high-pass filter consisting of a 10nF capacitor and 1kΩ resistor on the scope probe to block 50Hz, as shown in Fig.5. The narrow spike with its low repetition frequency defied photography on an analogue oscilloscope, so it is drawn in Fig.6.

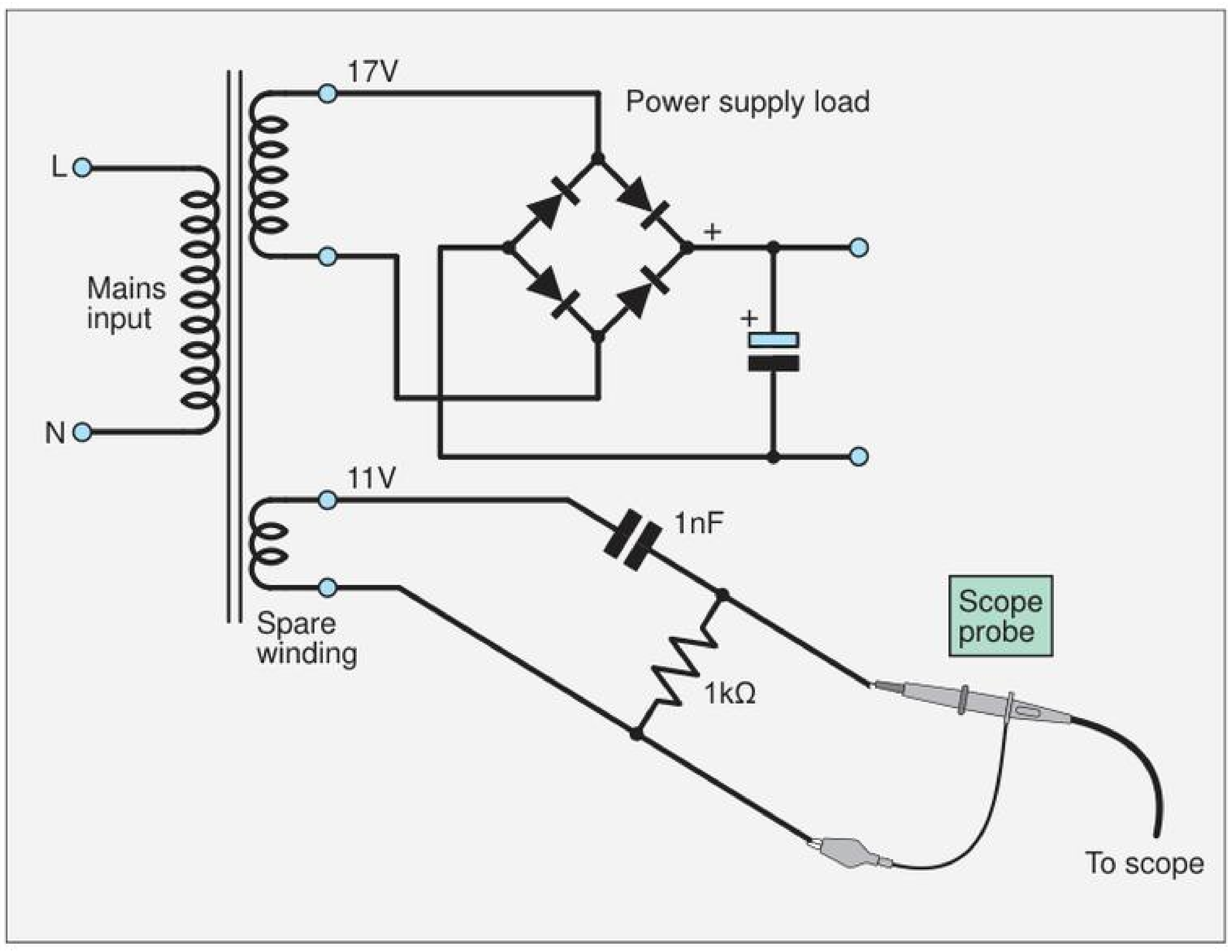

17V
Power supply load
L
Mains input
N
+
+
11V
1nF
Scope probe
Spare winding
1kΩ
To scope

Fig.7. An isolation transformer is normally needed to view switching spikes on bridge circuits. However, if the transformer has an extra secondary winding this can provide an isolated output.

I had to use an isolation transformer on the 'scope input to avoid earth currents with bridge rectifiers to look at the noise. It's easy if the transformer has dual secondary windings, since the scope can then be connected to a floating unused winding. In this way the mains transformer becomes its own isolation transformer, as shown in Fig.7.

Newer fast rectifiers, such as the UF4001 (the UF stands for ultra-fast), soft recovery, and Schottky diodes give a four-times smaller spike and consequently less RF noise. Snubbing capacitance is still needed, albeit a reduced amount, to reduce emissions further.

When it comes to suppressing these spikes, surprisingly cheap 'n' nasty capacitors can be very effective because they have high losses. So those much-derided barrier-layer ceramic disc capacitors from the Far East work very well. X7R and Y5V multi-layer ceramic types are also good. Plastic-film capacitors may need a series 2.2Ω to 100Ω resistor to provide a defined loss. The

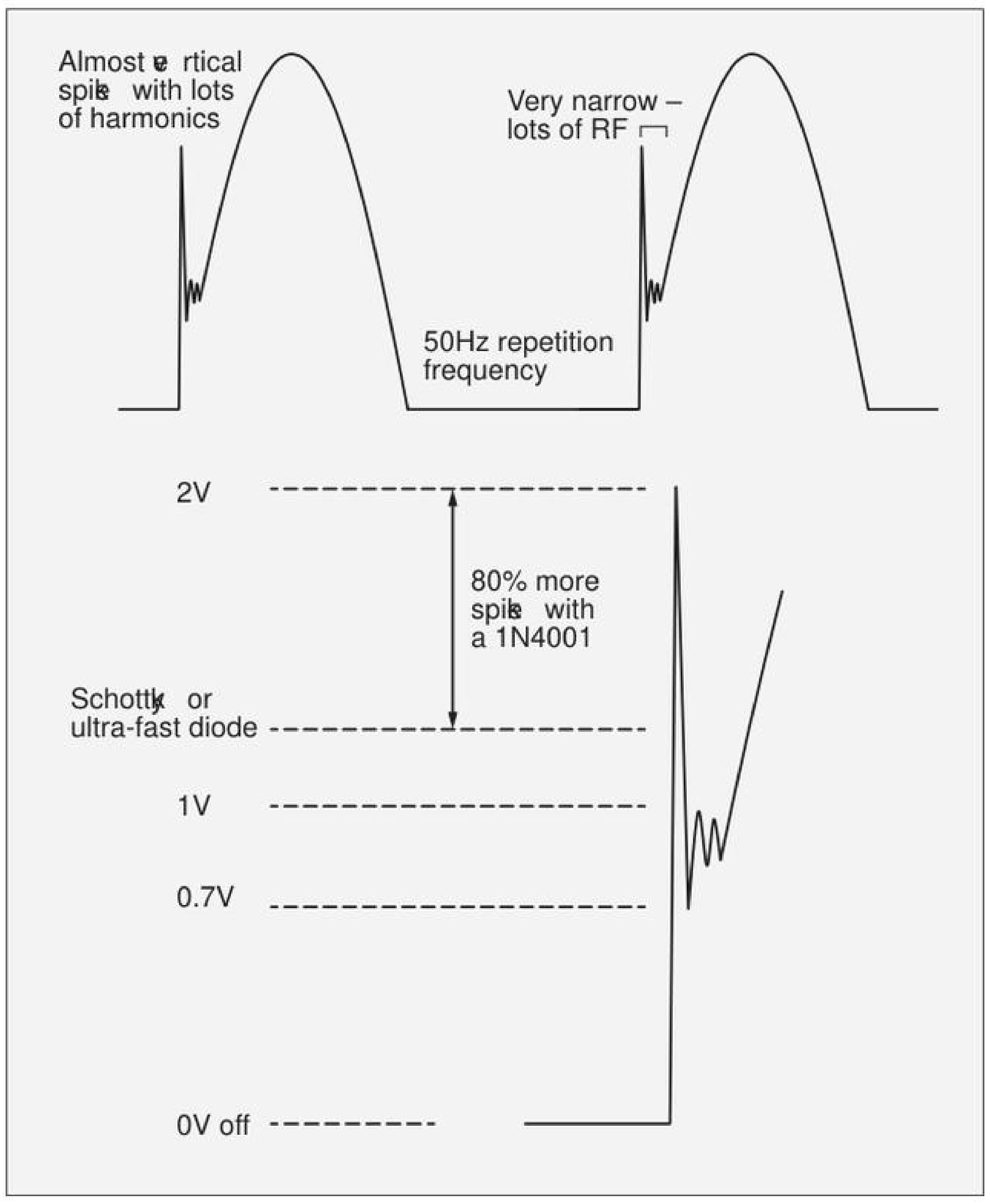

Almost vertical spike with lots of harmonics
Very narrow – lots of RF
50Hz repetition frequency
2V
80% more spike with a 1N4001
Schottky or ultra-fast diode
1V
0.7V
0V off

Fig.6. Illustration of rectifier switching spike.

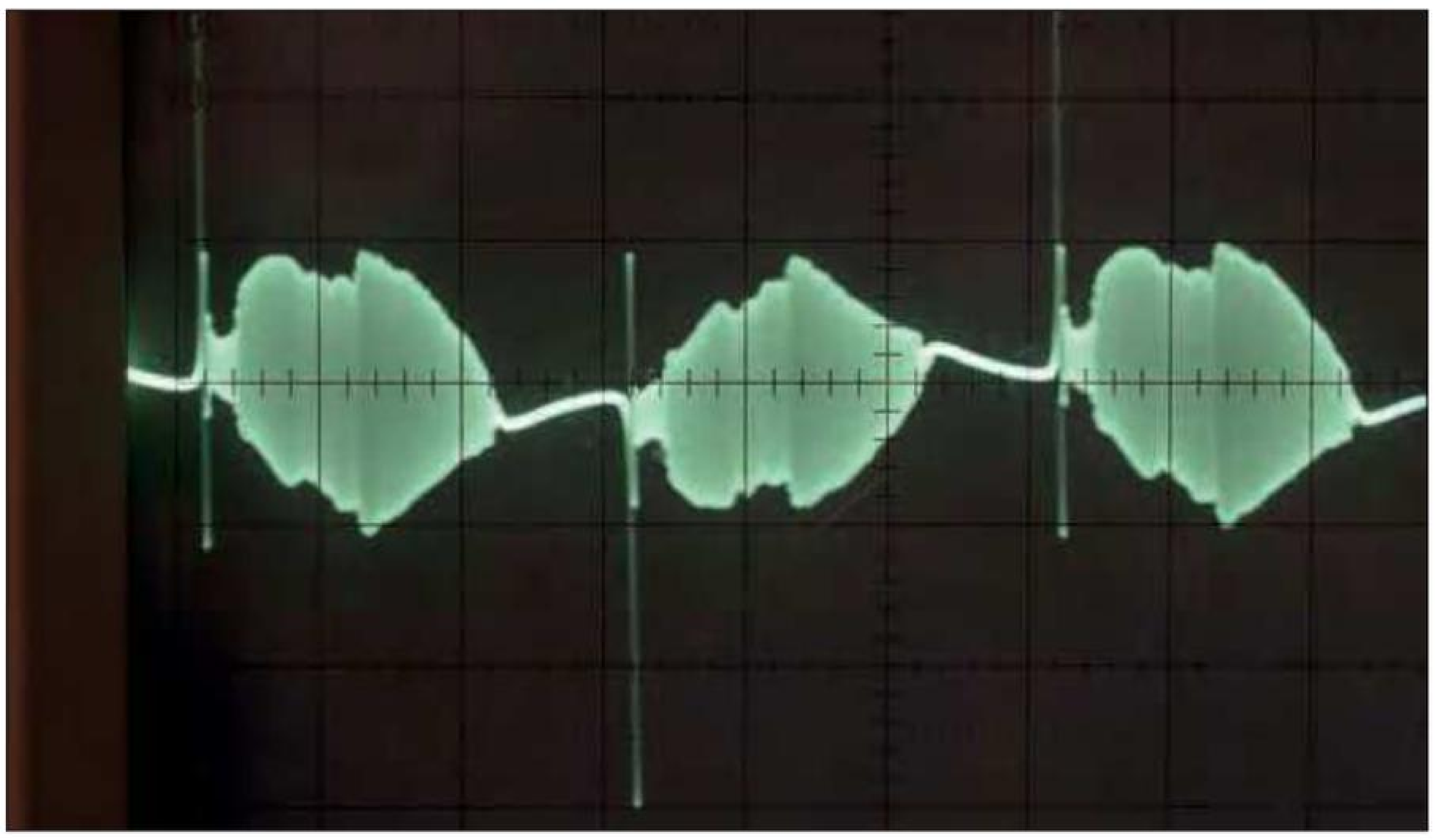


Fig.8. High-pass filtered waveform from a snubbed bridge rectifier circuit. Notice the resonant bursts and spikes. 400mV$_{pk\text{-}pk}$ (100mV/div) 100Hz repetition frequency .

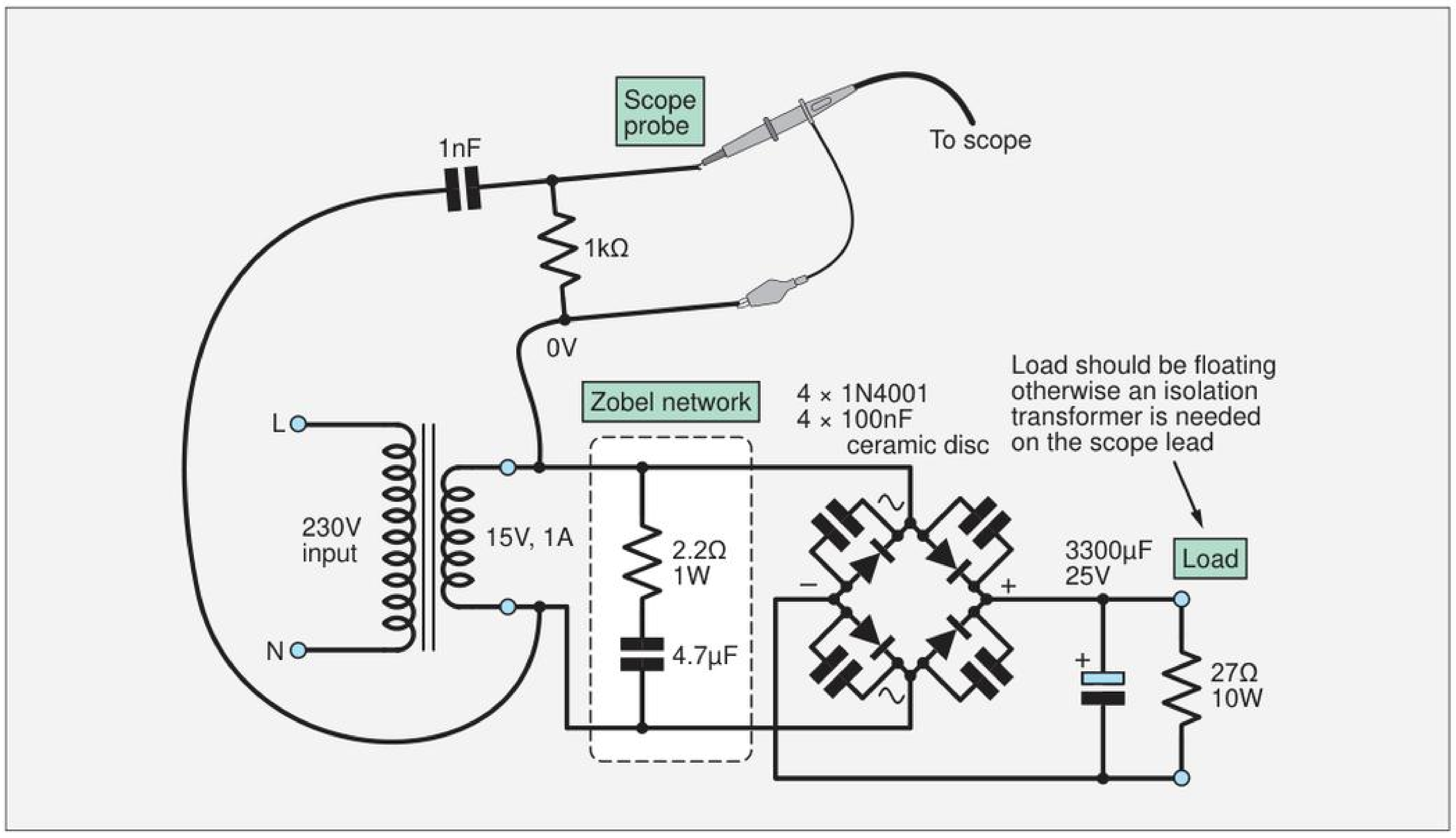


Fig.9. Adding a Zobel network across the transformer secondary.

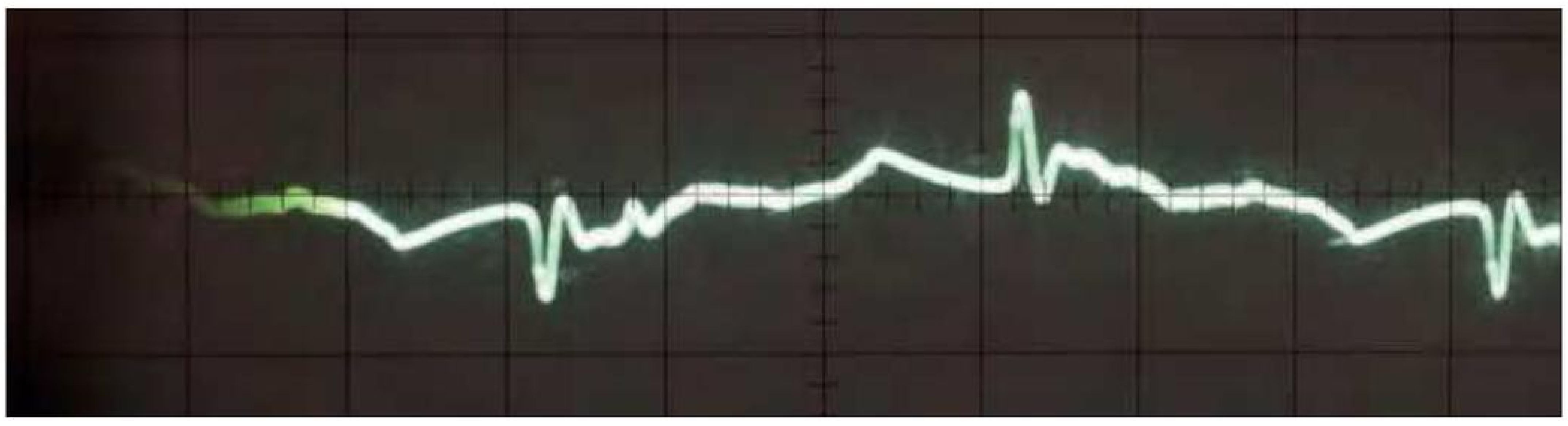


Fig.10. The effect of adding the Zobel network – the resonant bursts are damped down to 80mV$_{\text{pk-pk}}$ and the spikes are reduced.

capacitors should be wired across the diodes to minimise inductance and radiation loop area.

New buzz on the block

Diode switching also excites the resonance produced by the transformer's leakage inductance and winding capacitance (Fig.8). This is at a much lower frequency, typically 10kHz to 80kHz and can easily be suppressed by a Zobel (series RC) network across the secondary, shown in Fig.9. This does not affect radios and Theremins, but can affect audio if multiple power supplies are being used, causing a buzz due to beating between their respective frequencies. Typical values would be 1Ω to 100Ω and 1µF to 10µF.

Whacky circuit

Adding spike-snubbing capacitors without damping resistors boosts this transformer resonance. The effective damping produced by adding a Zobel network can be seen in Fig.10. On one transformer, I was pleased to find adding the Zobel also reduced the high frequency content of the mechanical hum. Using the transformer specified here, the Zobel was effective for both rails when put on one winding only. This is because the windings are magnetically closely coupled. A network of 2.2Ω and 10µF was found to be most effective when wired across the lower voltage 11V secondary. The capacitor has to be non-polarised and is therefore physically large. A non-polarised electrolytic can be used, such as those used in loudspeaker crossover networks. Since the current through this capacitor is quite large at 38mA, you might think this is a waste of energy. However, because the current is 90° out-of-phase with the voltage, the real power loss is quite small. The current can be put to good use to drive an LED, avoiding the heat produced from the normal dropper resistor, by using the circuit in Fig.11. Someone told me recently that this was a typical mad Jake circuit! I always like to include one in every column, otherwise you might as well be reading a textbook.

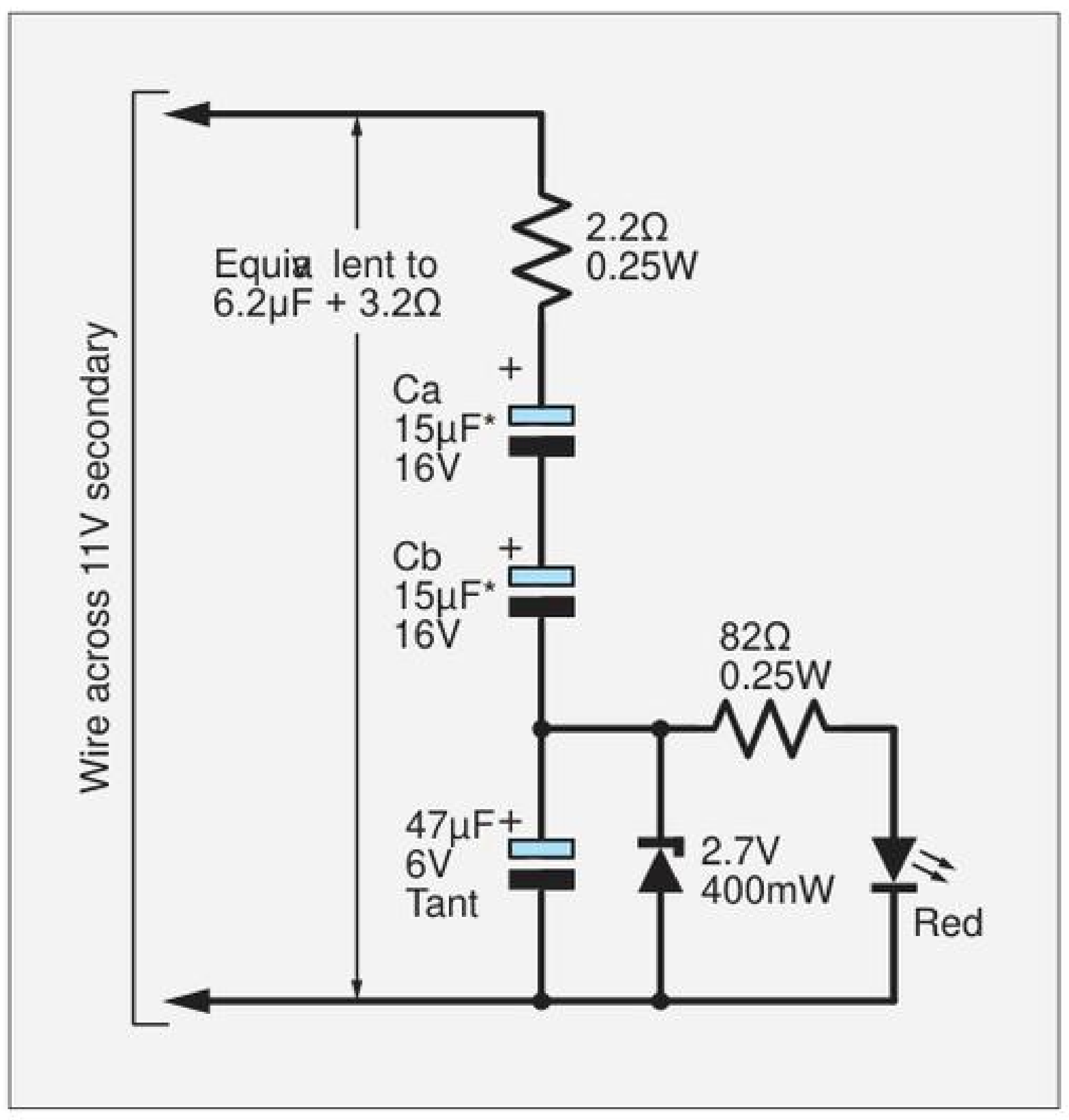


Fig.11. A whacky circuit? Incorporating an LED into the Zobel network. This avoids a heat-dissipating resistor.

This Zobel circuit is optional, since it is not strictly needed with the Theremin, but is worthwhile for studios where multiple power supplies are in use. A plastic film type is best, but a non-polarised capacitor can be made from two back-to-back electrolytic capacitors – Ca and Cb on Fig.11. Note the series connection reduces their value to half. Solid-aluminium types are the best choice here, since the high ripple current won't dry them out over time. A third capacitor C3, with six times the value of the non-polarised capacitor acts, as a capacitive potential divider. This develops a few volts across it to drive the LED. It also maintains the high frequency bypassing effect. A Zener diode stabilises the voltage at 2.7V to drive the LED via a current-limiting resistor. On negative cycles the Zener acts as a normal diode, clamping the reverse voltage to 0.7V. This means a normal polarised capacitor can be used. Since most of the current bypasses it, a tantalum type is usable. R1 sets the damping and limits any surge currents at switch-on.

Special offer transformer!

If there's one thing I detest it's the scrapping of perfectly sound components. I bought a whole load of mains transformers for scrap value that were being dumped because they had no built-in thermal cutout. I throw more transformers in the bin due to random thermal cut-out failures (where they are often embedded in the windings) than any other cause. I always insist on an external cut-out, where it can be replaced. The 'Right to Repair' movement is gaining ground and hopefully practices like embedded cut-outs will be banned. There will be no shortage of these transformers, because PCB designer Mike Grindle and I have over 200 in stock. Even if we do run out, Mike will quickly edit the PCB for a new transformer.

The transformer used has two secondaries of different voltages, as shown in Fig.12. Thus, the board has provision for two separate power supply circuits. Alternatively, the two secondaries can be wired in series to give a higher

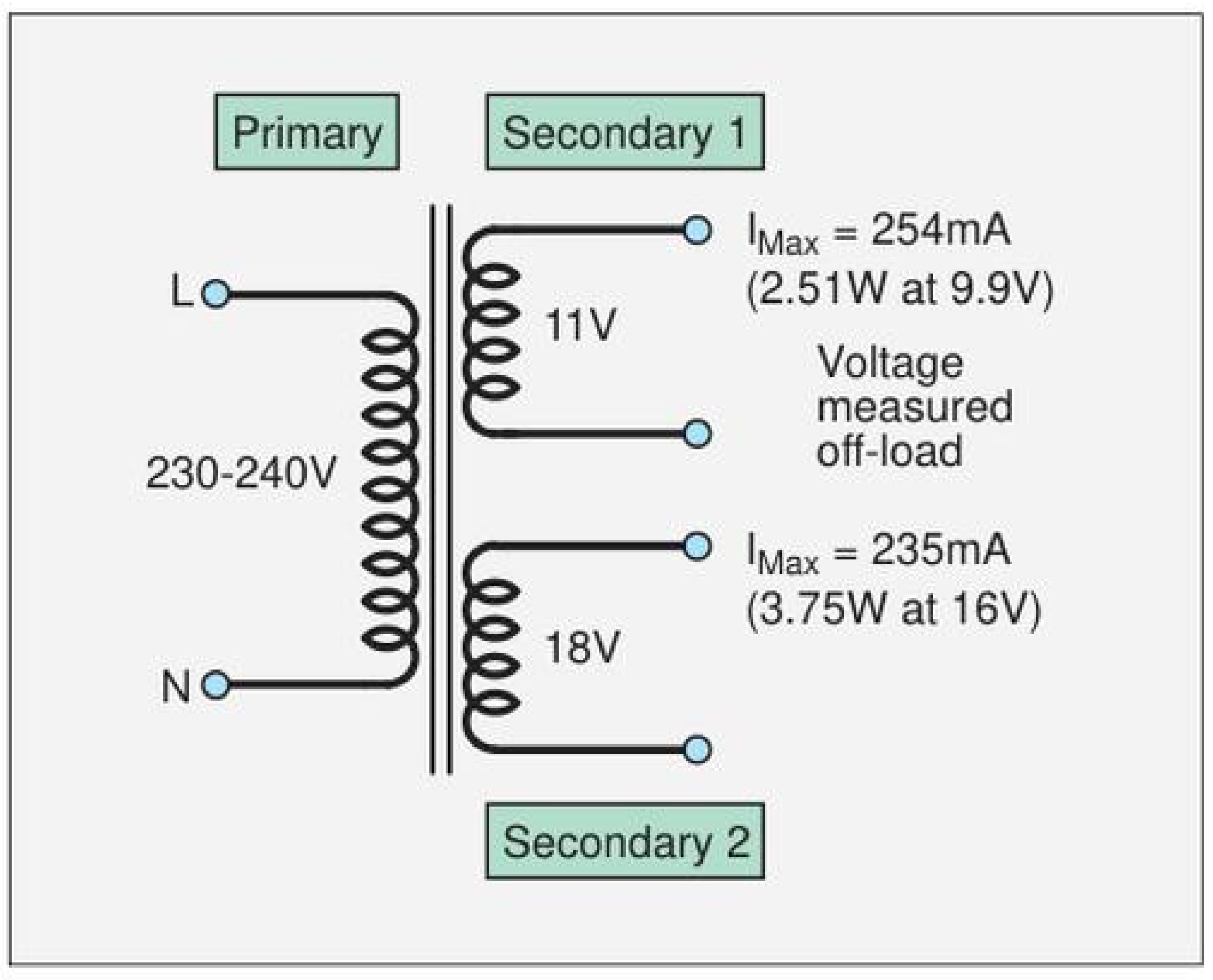

Primary
Secondary 1
L
11V
I$_{Max}$ = 254mA (2.51W at 9.9V)
Voltage measured off-load
230-240V
I$_{Max}$ = 235mA (3.75W at 16V)
18V
N
Secondary 2

Fig.12. The transformer specified has two secondaries.

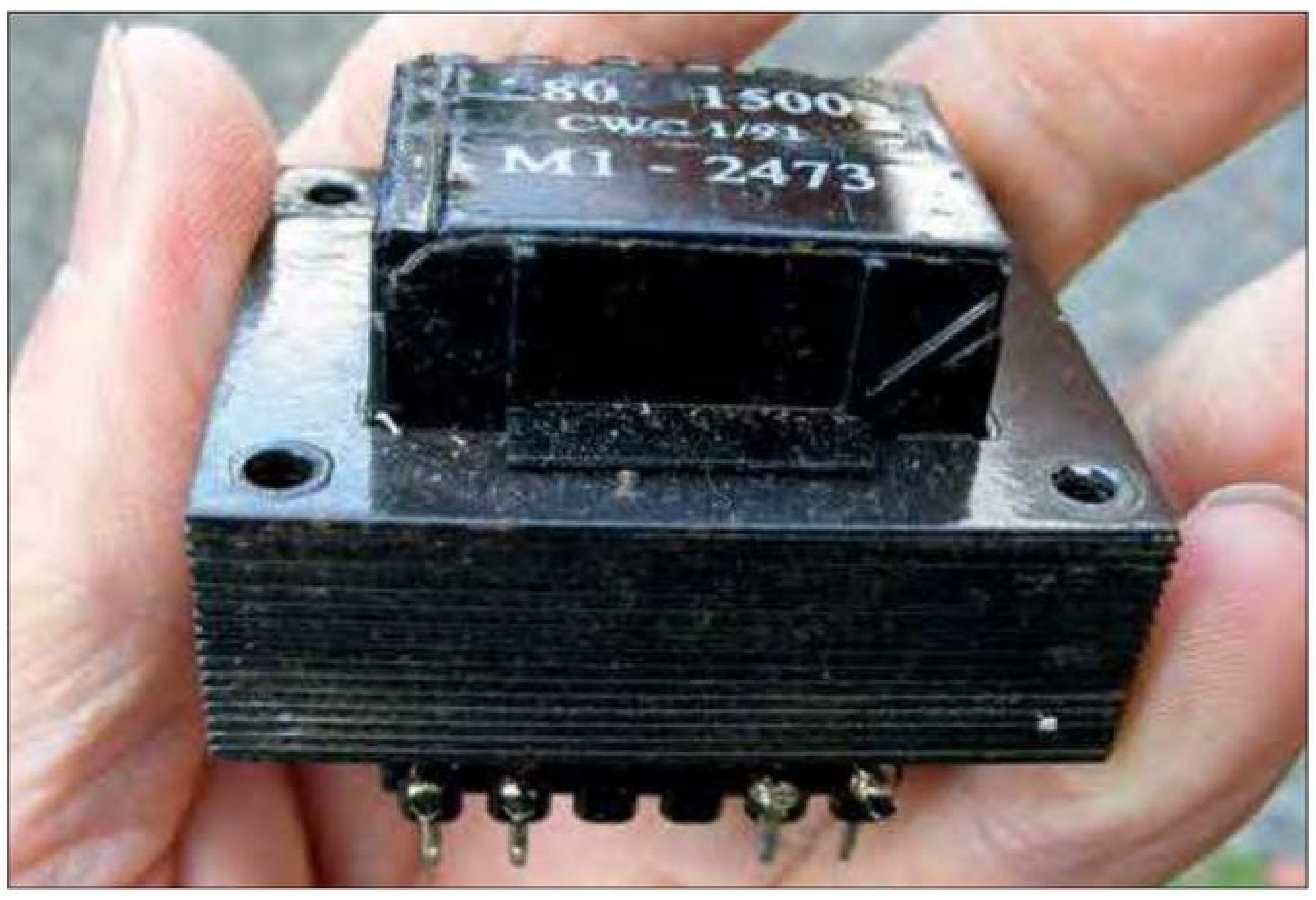


Fig.13. These excellent transformers were going to be dumped – nothing wrong with them and they are ideal for our Theremin power supply.

voltage single supply. The transformer is shown in Fig.13.

It is possible to connect the two positive supplies to give a dual-rail plus-and-minus supply with a centre ground, as shown in Fig.14. This is not as good as a proper dual-rail supply with a negative voltage regulator. The 0V reference can bounce around if too much current goes from the 'negative rail' into ground. This is because the output impedance of the regulator is in series with the ground line. This was how early op amp circuitry was originally powered; we just designed it so current was never dumped into ground. It was always arranged to go from one rail to the other. This is still good practice today, unless your ground is a superconductor.

Transformer specification

When using surplus components with no written specification (such as this one), it is important to take some measurements. The first aspect to consider is the transformer's physical size, which will give some indication of the power rating. This is usually specified in terms of VA for transformers rather than watts because the current is pulsed when feeding a capacitor-smoothed rectifier. Most small mains transformers are built up from standard lamination sizes, so by looking at the Danbury and Vigortronix catalogues I determined it was about 8VA. Small transformers like this generally have a regulation figure (how much the voltage drops on full load current) of 22%, so the current rating of the secondaries can be determined by loading with big wire-wound resistors until the expected voltage drop is reached. Finally, the transformer should be left on for a long time at estimated full load for a while to make sure it does not get too hot. Above around 70°C is too hot; if a smell of burning polyurethane varnish fills the air, it is definitely time to turn it off.

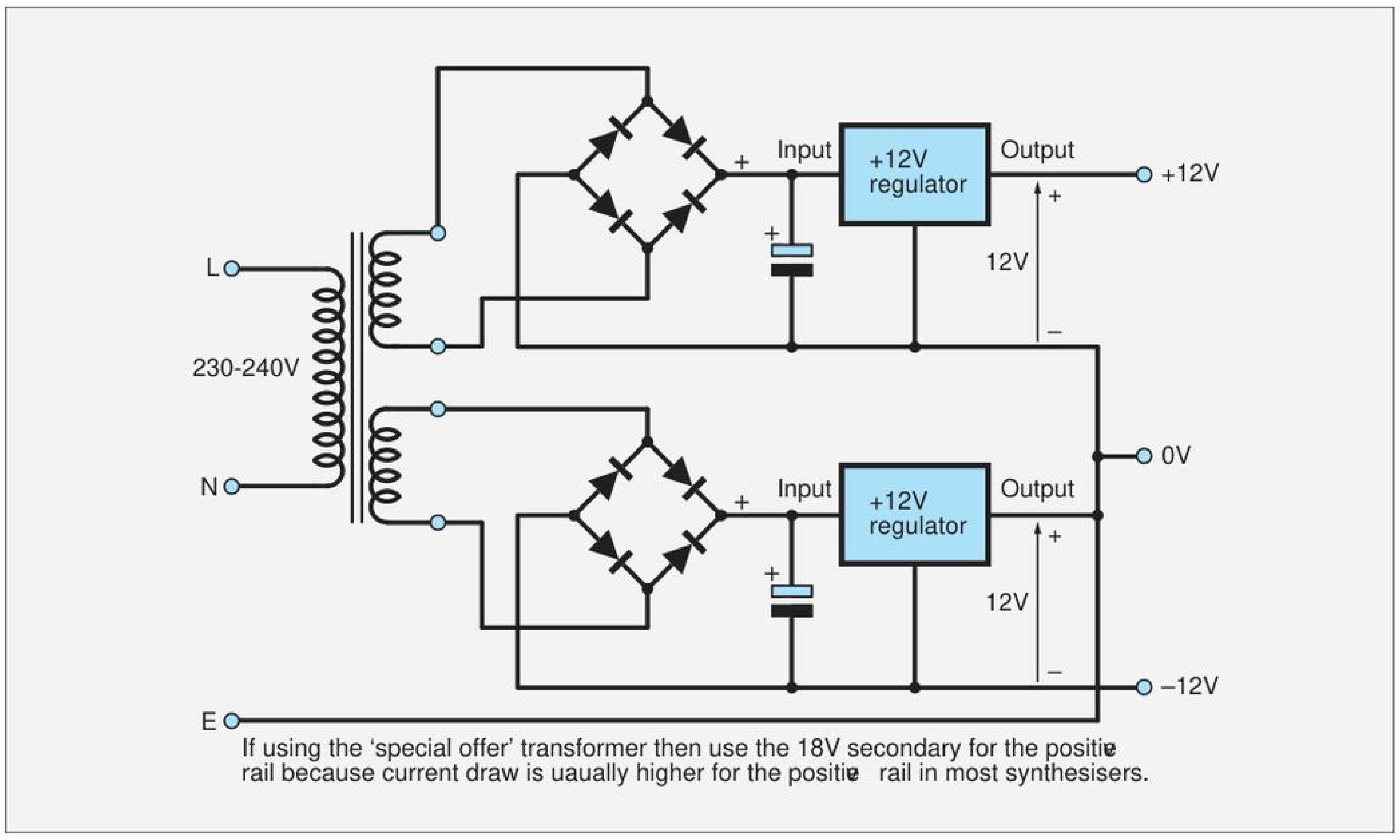
Transformer measured specification

Size (mm)	47 × 36 × 40 (w × l × h)
Weight (g)	250
Off-load voltages	18V and 11V.
On-load voltages (at max current):	
	16V @ 235mA, load 68Ω
	9.9V 254mA, load 39Ω.

With the secondaries in series, 26V was obtained with a load of 240mA, temp rise was 45°C above ambient.

Next month

That wraps it up for this month. As you can see, even simple power supplies can be complicated, or rather they need care and consideration at the design stage. Next month, we'll build the circuit.



If using the 'special offer' transformer then use the 18V secondary for the positive rail because current draw is uaually higher for the positive rail in most synthesisers.

Fig.14. Connecting two positive regulators to make a plus/minus power supply.

PoLabs

For more information or software download please visit

www.poscope.com/epe

PoKeys
Connect, Control

- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC

Stepper motor drivers
Drive

- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A

PoScope Mega1+
PoScope Mega50
Measure

- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

Make it with Micromite

Phil Boyce – hands on with the mighty PIC-powered, BASIC microcontroller

Part 19: Controlling your MRB with an IR transmitter and animating its eyes

The two LED 8×8 matrix modules added last month provide the Micromite buggy with a nice-looking pair of eyes. They enable you to give your robot some personality, especially when they are animated. For example, why not make the eyes appear to look left when the robot is turning left, or possibly make them blink at random intervals. Maybe the eyes could close when the robot has not moved for a while, giving the appearance of it snoozing. They could then open immediately before the robot does anything, making it look as if it has just woken up.

An MMBasic program has been written to make all the above possible; so this month we are going to explore the detail of how to use the code. By the way, why not use these eyes as the basis for some other fun projects. For example, add a PIR sensor to create a spooky Halloween ghost that appears to stare at you as you move around the room!

This month, we will also discuss how to provide basic control of your robot with an infrared (IR) remote transmitter. We will use the 44-button IR transmitter from the *Micromite Moodlight* in *Part 13* (*PE*, February 2020). This will allow you to move the robot forward and back, turn it left and right, and also adjust its speed. Naturally, these actions will be combined with eye animations.

MRB Chassis

Before we go any further, several readers have asked for the *MRB* chassis dimensions so they can make their own – all is revealed in Fig.31 (see end of article). If you have access to a laser cutter or CNC machine then you will find these measurements will give you a perfectly accurate result. Be sure to use a material that is from 3mm to 6mm in thickness. Acrylic or modelling plywood is perfect; you could use aluminium, but in this scenario it should be painted (at least two coats) to avoid potential shorts with the underside of the *MRB* daughterboard.

If you intend to cut the chassis by hand, then the hole positions for the wheel mounts and motor mounts are critical (as are the hole diameters). If they are positioned too far apart, then the silicone-tracks will be too tight. This will then apply sideways tension to the motor spindles resulting in the motors being over-worked (shortening their life-span). On the other hand, if the wheel and motor mounts are positioned too close together, then the silicone tracks will be too loose and will keep falling off the toothed wheels they are wrapped around.

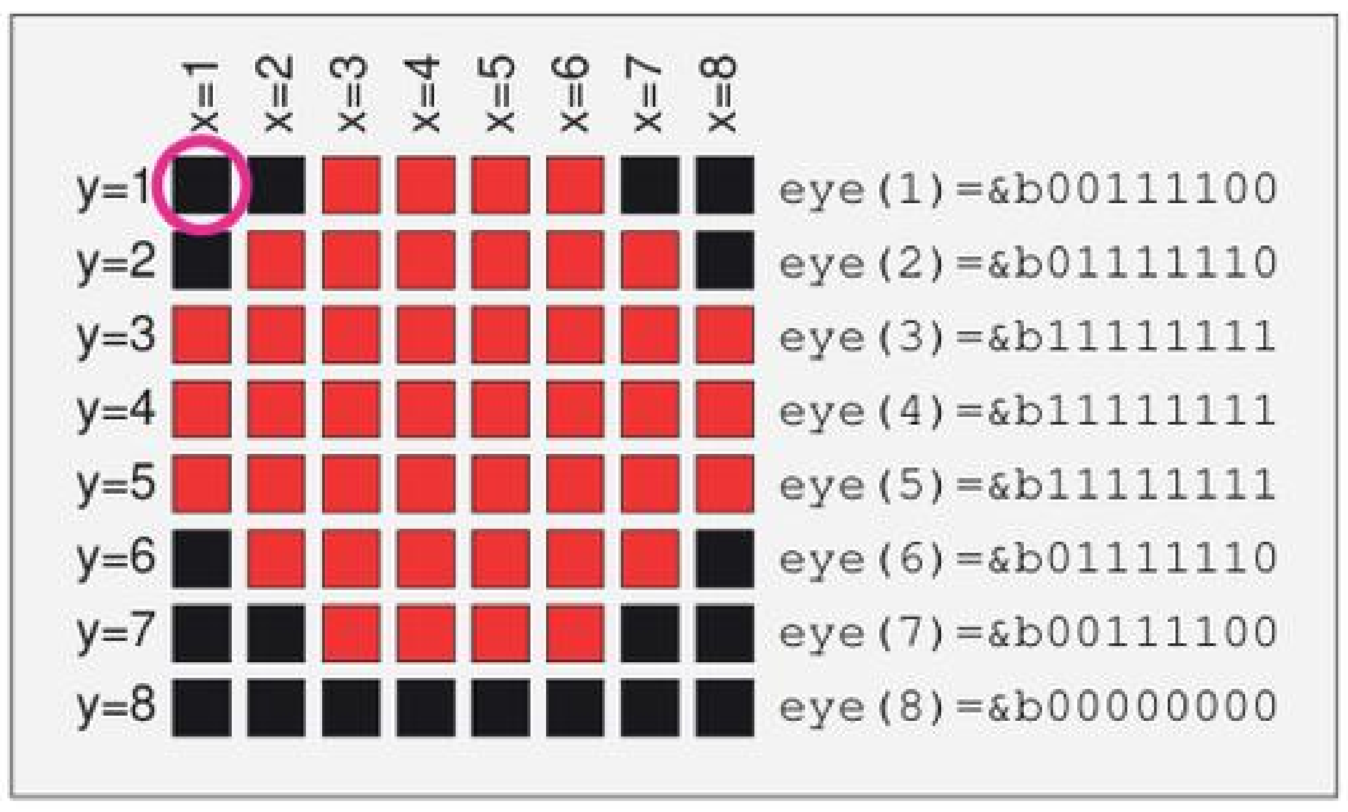
For the three slots for the *MKC*/Bluetooth module, and the two slots for the motors; there is a fair bit of tolerance in the size and positions of all of these, so cutting by hand shouldn't be too much of an issue.

If you do make your own chassis, then please do send in pictures – we are always interested in seeing how you get on. Thanks to all of you who have done just that – there are some great *MRB* builds out there.

Animated eyes – theory

If you loaded last month's test program then you have probably taken a quick look at the code to try and understand how the animated eyes work. If not, then don't worry – all will be revealed below. Even if you did look at the code, then this month we have added some new functionality; so let's work through what is involved, step-by-step. It is a lot easier than you may imagine. A summary of the individual steps is as follows:

- Define eyeball shape (8×8)
- Overlay pupil pattern (2×2)
- Overlay any blinking (by row(s))
- Display result on LEDs (anywhere)



```
eye(1)=&b00111100
eye(2)=&b01111110
eye(3)=&b11111111
eye(4)=&b11111111
eye(5)=&b11111111
eye(6)=&b01111110
eye(7)=&b00111100
eye(8)=&b00000000
```

Fig.32. The values in the **`Leye()`** and **`Reye()`** arrays determine the shape of the two solid eyeballs.

Note that both eyes (left and right) are independently controllable. This allows for a lot more character to be added compared to if we were only able to display the same image on both eyes. All variables are prefixed with either an `L` or an `R` (you can work out the logic there!) – and any variable in the following discussion without an `L` or an `R` prefix is just to make the diagrams and explanations less congested.

The eyeball

The whole animated eye process begins by defining the shape of each eyeball; ie, the solid 'white' part of the eye. In our application, the eyeball will be a solid colour of whatever LED matrix colour you have – here we are using red.

Typically, the two eyeballs are either identical (symmetrical), or are a 'mirror' image of each other (but, this does not have to be the case). In Fig.32 you can see that the eyeball is a pattern no bigger than 8×8 pixels. Here we have defined an eyeball that is 8 pixels wide, and 7

Micromite code

The code in this article is available for download from the *PE* website.

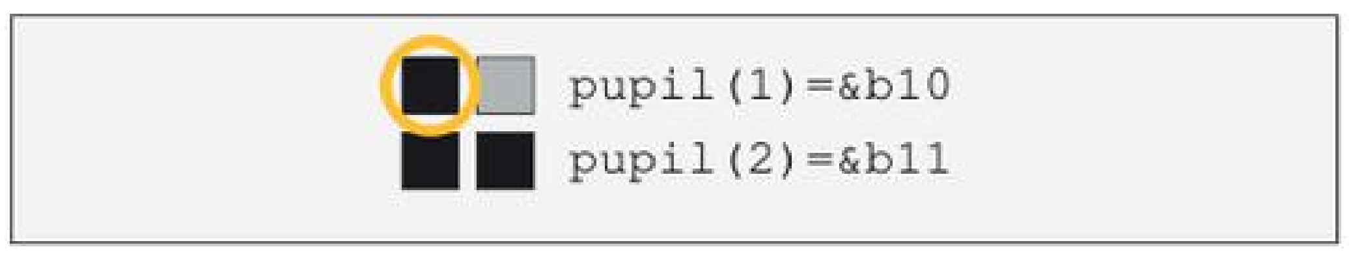


Fig.33. The values in the **`Lpupil()`** and **`Rpupil()`** arrays define how each 2×2 pupil will look. There are 16 different permutations for each pupil.

high. This pattern is stored in an array named `eye()`. So `Leye(1)` contains the required pixel pattern of the top row of the left eye, and `Reye(8)` stores the pixel pattern of the bottom row of the right eye. The 8 bits that define the pixel pattern in a horizontal row translate nicely into an 8-bit binary value and hence we are using the MMBASIC `&b` binary prefix to show how these values are easily generated from the required pattern (`1` represents ON, and `0` represents OFF). An important point to note here is that each 8×8 eyeball pattern is *not* defining the ON/OFF LEDs on a matrix, it is just the 8×8 eyeball pixel image that we will ultimately be able to position anywhere on the 8×8 matrix (as we shall see in the last step).

So this step has simply defined the eyeball shape as a set of ON pixels, stored in the `Leye()` and `Reye()` arrays. Behind the scenes in the code, the `eye()` arrays are stored into temporary arrays: `LeyeTemp()` and `ReyeTemp()`. These two temporary arrays allow the overall eye images to be 'constructed'.

The pupil

Referring to Fig.33, you will see that the eye's 'pupil' is a 2×2 block of pixels. Again, we will use the `&b` binary prefix to make things easier to understand. The concept here is to use OFF pixels to represent the pupil shape. The binary notation shown here uses a 1 to represent a dark part of pupil (ie, pixel off) and a `0` to represent 'no change' to the background. This allows us to have any shape pupil in the 2×2 block (yes, that's 16 different pupil shapes if you do the maths!). Here we are showing a pupil shape with the top-right corner not darkened.

You may be wondering why we are saying 'no change' rather than simply turning ON the pixel (bearing in mind the eyeball comprises of pixels that are ON). Well, the pupil can be placed anywhere over the eyeball image, and if it were to be placed next to the edge of the eyeball (great for 'sad' eye impressions) then we don't want to affect the shape of the eyeball by turning ON 'un-darkened' pupil pixels that hang outside the eyeball shape.

So `pupil(1)` contains the top row of the 2×2 block, and `pupil(2)` contains the bottom row. A `1` represents a dark pixel in the pupil, and a `0` represents no change.

Now that we have defined the pupil shapes, we need to position them within the eyeball image (stored in `LeyeTemp()` and `ReyeTemp()` – see above). This is just a matter of defining where the top-left pupil pixel is positioned within the eyeball 8×8 image by using a simple *x*,*y* co-ordinate system. The top left corner of the eyeball is defined as *x*=1, *y*=1. So `Lx=4` and `Ly=4` would position the 2×2 left pupil in the central position of the 8×8 left eyeball. Similarly, `Rx=3`, `Ry=5` would position the right pupil near the lower-left corner of the right eyeball.

So this step has simply defined the pupil shapes as a set of OFF pixels stored in the `Lpupil()` and `Rpupil()` arrays; and positioned both pupils into the correct positions within `LeyeTemp()` and `ReyeTemp()` by defining `Lx` and `Ly`, and `Rx` and `Ry` values.

Blinking

First, a bit of background. We are currently working through how to construct a static image of two eyes to be displayed on the two LED matrix modules. Consider the result as a 'frame' image. If we were to display one frame, followed in quick succession by another frame (that has a slight difference in content), then continuing this process we would end up with animated eyes – you can liken this to the early days of film animation.

So a closing-eye effect can be generated by effectively switching OFF appropriate rows of LEDs in each frame and displaying them in quick succession. However, you can't simply switch the LEDs back ON again to simulate an opening-eye effect otherwise you will end up with all LEDs being turned ON.

By using `LeyeTemp()` and `ReyeTemp()` each and every time we draw a frame, we can overcome this opening-eye issue. The eyeball shape, and pupil shape, are stored in the `eyeball()` and `pupil()` arrays, and the image is constructed in the `eyeTemp()` arrays (piece by piece) to create the single 'frame'. To create a single frame within a blinking effect (sequence of frames), we ideally need a method to turn off the pixels in any row(s) in the current `eyeTemp()` arrays (currently containing the eyeball, and correctly positioned pupil). We could have simply defined new `eye()` arrays, but this would destroy the original eyeball pattern stored. We have a better method as we will now see.

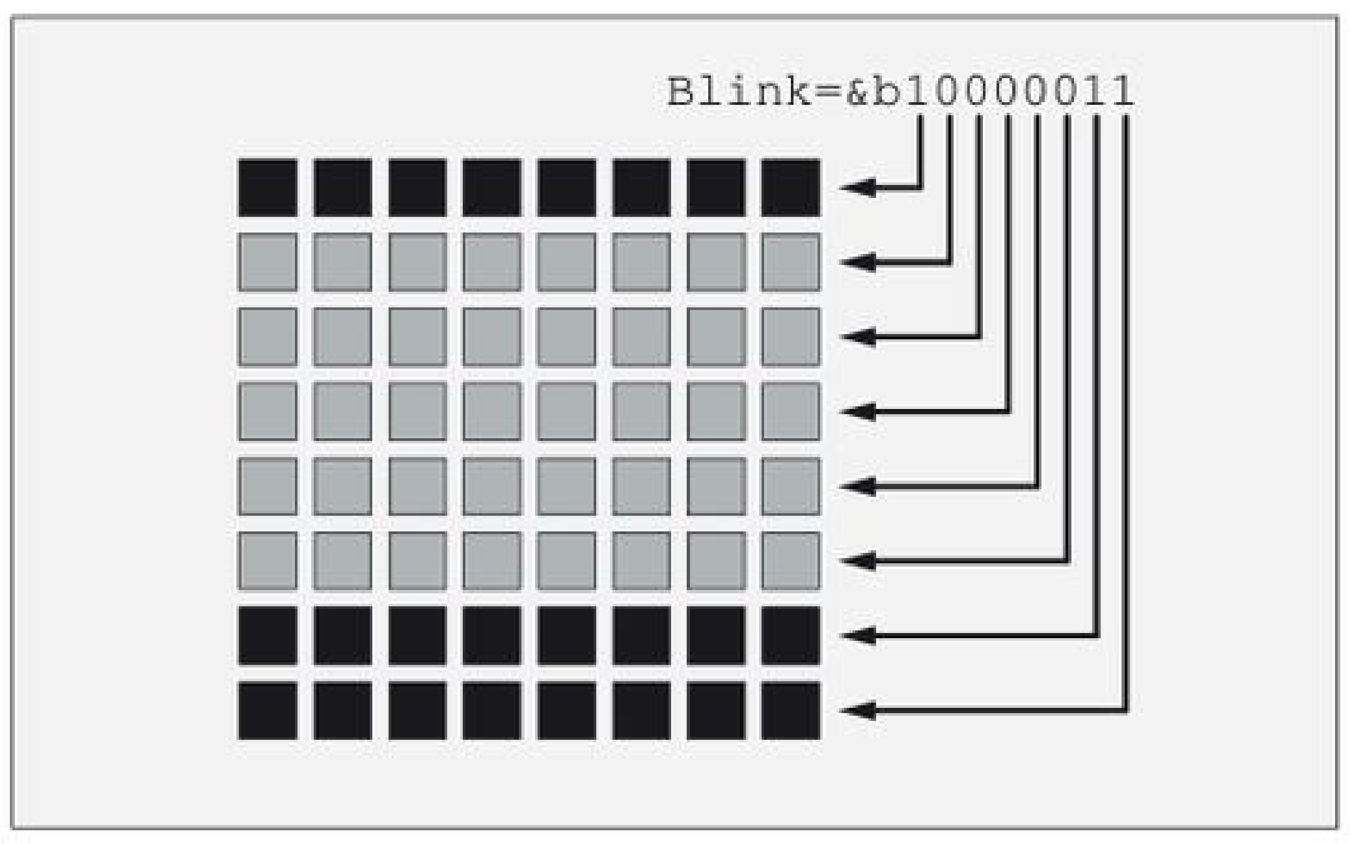


Fig.34. The values in **`Blink`** determine which row(s) of pixels in the eye image need to be switched off. Here, the top and bottom rows are switched off in our 8×7 eyeball.

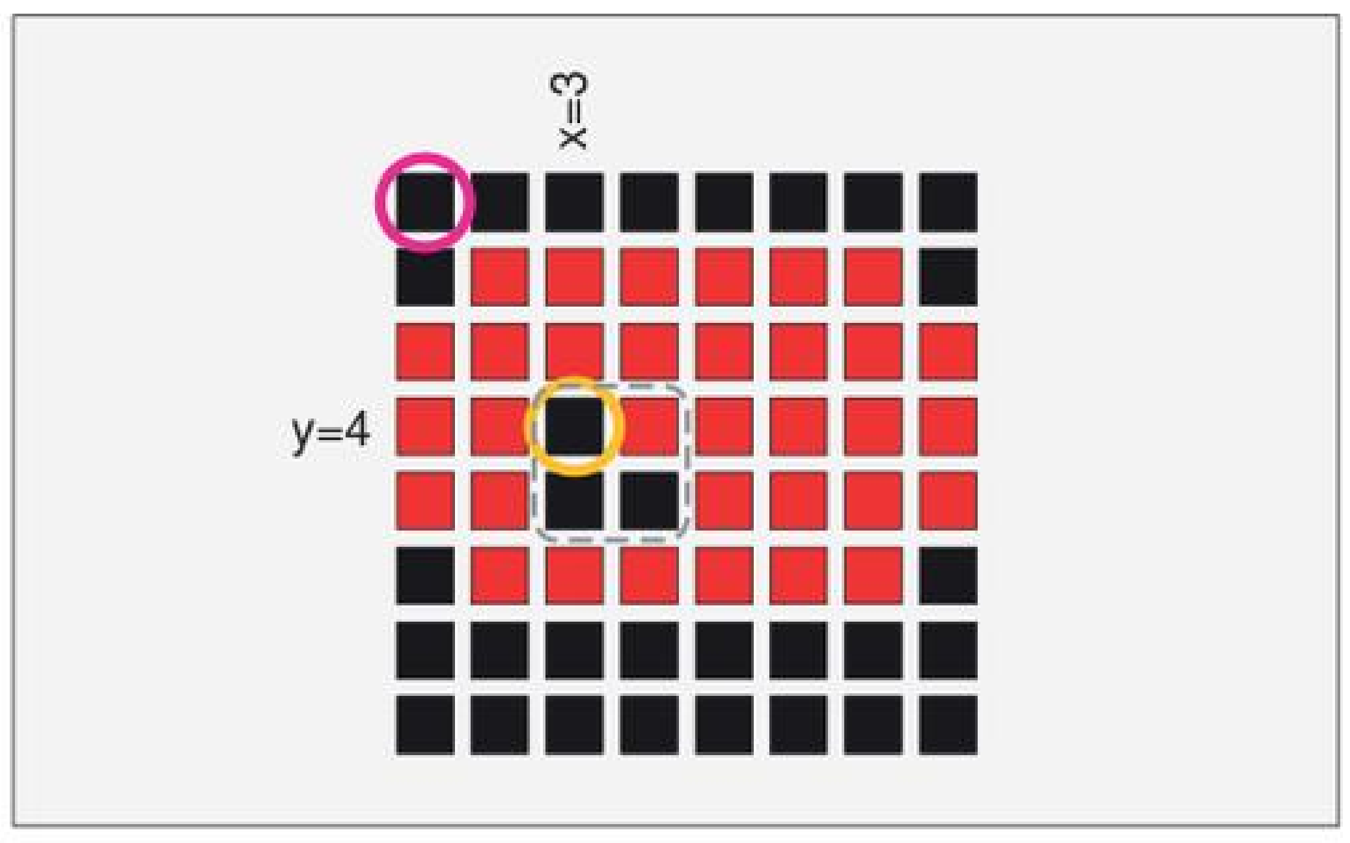


Fig.35. The eye image after defining the eyeball, the pupil, and any **`Blink`** setting. Values of **`Lx`**/**`Rx=3`** and **`Ly`**/**`Ry=4`** define the pupil position (shown highlighted).

Referring to Fig.34 you will see that the `Blink` variable is used to store a value to represent any complete row(s) of pixels that we wish to turn OFF in the `eyeTemp()` arrays. Once again, binary notation is used to simplify the explanation. A `0` bit in `Blink` means that the associated row is unaffected (and will be displayed as currently defined in `eyeTemp()`, and a `1` means that a row will be switched OFF in the displayed image. The 8-bit value contained in `Blink` uses one bit per row of pixels – the most-significant bit relates to the top row, and the least-significant bit relates to the bottom row. Note once again, this is not referring to the rows on the matrix module, it is referring to the built up image in the `eyeTemp()` arrays (which can ultimately be positioned anywhere on the LED matrix modules). Here we are effectively switching OFF the top and bottom rows of our 8×7 eyeball image.

So this step has simply defined which rows of pixels we wish to switch OFF in order to create a frame within a sequence of frames that can be used to create a blinking effect.

At this stage it is worth seeing the result stored in the `eyeTemp()` arrays from the values shown in Fig.32, 33, and 34. Referring to Fig.35, you can see the result of using *x*=3 and *y*=4 as the pupil position values. Here, the (circled) pixel at 3,4 is defining the top-left pixel of the pupil.

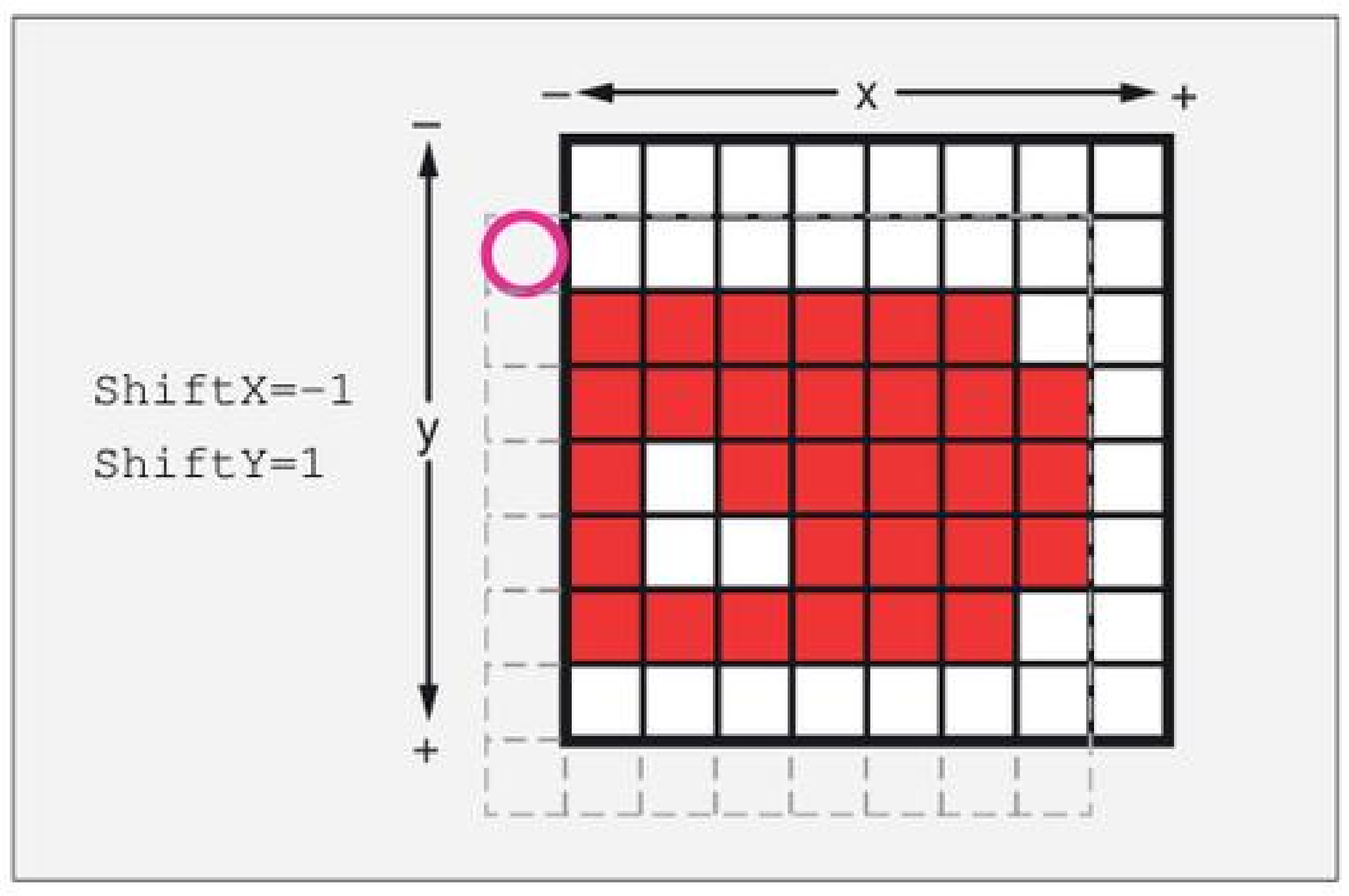


Fig.36. The **`LShiftX`** / **`LShiftY`**, and **`RShiftX`** / **`RShiftY`** variables determine where the eye image is positioned on each LED matrix. Here they are moved one pixel to the left, and one down.

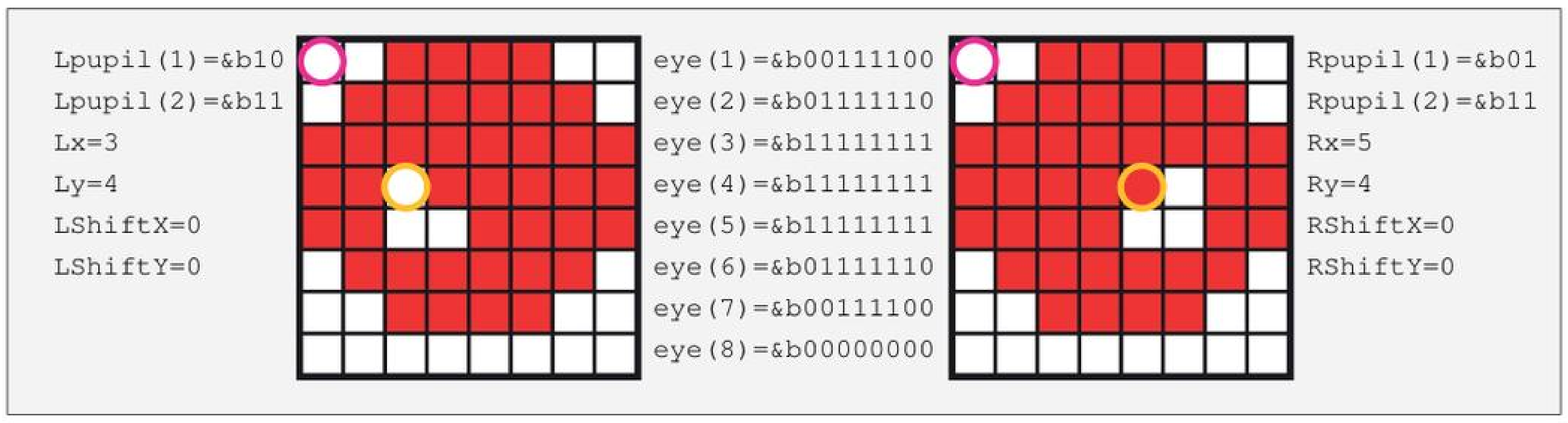

Lpupil(1)=&b10
Lpupil(2)=&b11
Lx=3
Ly=4
LShiftX=0
LShiftY=0
eye(1)=&b00111100
eye(2)=&b01111110
eye(3)=&b11111111
eye(4)=&b11111111
eye(5)=&b11111111
eye(6)=&b01111110
eye(7)=&b00111100
eye(8)=&b00000000
Rpupil(1)=&b01
Rpupil(2)=&b11
Rx=5
Ry=4
RShiftX=0
RShiftY=0

Fig.37.The demo program (**MRB_SampleEyes.txt**) is configured to output the image shown here. Try altering variables in the code to see how they affect the displayed image.

The other circled pixel at 1,1 is used to define where this 8×8 image is positioned on the 8×8 matrix module (explained in the next step).

Displaying the eyes

As mentioned several times already, the image contained in the `eyeTemp()` arrays can be positioned anywhere on the LED matrix. This is where two more variables come into play for each eye: `ShiftX` and `ShiftY`. Referring to Fig.36, you can see how by setting these values, you can place the image wherever you like on the LED matrix modules. Here we have `ShiftX=–1`, which means the image is positioned 1 pixel over to the left; and `ShiftY=1`, which means it is also positioned one pixel down. The dotted line in Fig.36 represents the 8×8 image stored in the `eyeTemp()` arrays with its top-left pixel circled. `ShiftX` and `ShiftY` simply give coordinates relative to the top-left pixel on the LED matrix module. Note that values bigger than `8`, or smaller than `–8` will make the image disappear.

Now that `ShiftX` and `ShiftY` have been defined, we need to call a subroutine to actually draw the eyes onto the LED matrix modules. This subroutine is simply called `DrawEye` and will work out all the logic in order to create the desired result. By altering the `ShiftX` and/or the `ShiftY` values, the eyes can be made to scroll. Note that there is no wrap-around when scrolling.

A quick example

To demonstrate all of the above, we will now draw a pair of simple eyes on the robot (ie, draw a single frame). This is easy; it's really just a matter of loading the variables and arrays, and then calling the `DrawEye` subroutine. All the information required is shown in Fig.37; and you can download the code (**MRB_SampleEyes.txt**) from the August 2020 page of the *PE* website.

To begin with, the `Leye(1)` to `Leye(8)` and `Reye(1)` to `Reye(8)` arrays are loaded with the eyeball pattern as shown. Next, the pupils are defined by loading arrays `Lpupil(1)` and `Lpupil(2)`, and `Rpupil(1)` and `Rpupil(2)`. To position the pupils, load the variables `Lx`, `Ly` and `Rx`, `Ry`. Here, we are not setting `LBlink` or `RBlink` (but you can set them to `0` for now so you can see the effect later). To finish defining the 'frame', set `LShiftX` and `LShiftY`, and `RShiftX` and `RShiftY` as shown. Nothing will appear on the LEDs until you call the subroutine `DrawEye` – so go ahead and do this to see the result. If all has been typed in correctly, you should see the image (as shown in Fig.37) displayed on the robot's eyes.

Now have a play by altering some of the above values and see how it affects what is displayed on the LEDs. This is the best way to learn. If you look at the code, there are also some `DO…LOOP`s contained in subroutines, which when called, will show a sequence of frames – in other words, will show you some eye animations. Now have a go at creating your own animations!

IR control – theory

Last month, the *MRB* had IR functionality added by implementing just a single component – the now-familiar TSOP IR receiver. As we have seen in previous articles, MMBASIC makes it very easy to add IR control into program code. As a reminder, we use the `IR` command to define an interrupt subroutine (`SUB`) that will then automatically get called whenever an IR signal is detected. MMBASIC passes two variables into the interrupt subroutine: the `KeyCode` (representing the unique button number that has been pressed), and the `DeviceCode` (representing which IR transmitter is being used).

Within the IR interrupt subroutine, we can simply use the command structure:

```
SELECT CASE KeyCode … CASE x …
CASE y … CASE z… END SELECT
```

where `x`, `y`, and `z` are the values of specific buttons on the IR transmitter that we wish to respond to. So in order to make the robot do something useful on specific button presses, we will need to write some code for each `CASE`.

As with any interrupt subroutine, it is good practice that each `CASE` in the IR subroutine contains minimal code. That way, the time spent in the interrupt routine is minimised, and hence the program will always be able to respond to other interrupts – in this case, immediately respond to other button presses. The best way to minimise code is for each `CASE` to simply alter the values of certain variables; variables that you are using (and continually checking) in your main program. We will now show you this in practice.

Bringing it all together

Now that you have an understanding of how to create, draw, and animate some eye effects, let us quickly show you how to apply them to some robot movements – all controlled by the 44-button IR remote control.

The program code we'll be using (**MRB_IR_Control.txt**) is available for download from the August 2020 page of the *PE* website. We are not going to go into all the specific detail here because the code is commented throughout. Instead, we want to provide you with an overview of the techniques being used. There is nothing new, and a lot of the topics have already been covered. I hope the code will inspire you to create your own much better program, which is customised to your own needs.

Fig.38 provides an overview of the code. It has four sections, with which you should now be familiar:

- Set up
- Main program
- Subroutines
- Interrupt subroutines.

The Set up code just sets a few things that need setting – take a look at the code and I promise that if you've followed this series then it will all make sense.

The Main program comprises of a `DO…LOOP` and works through two main blocks

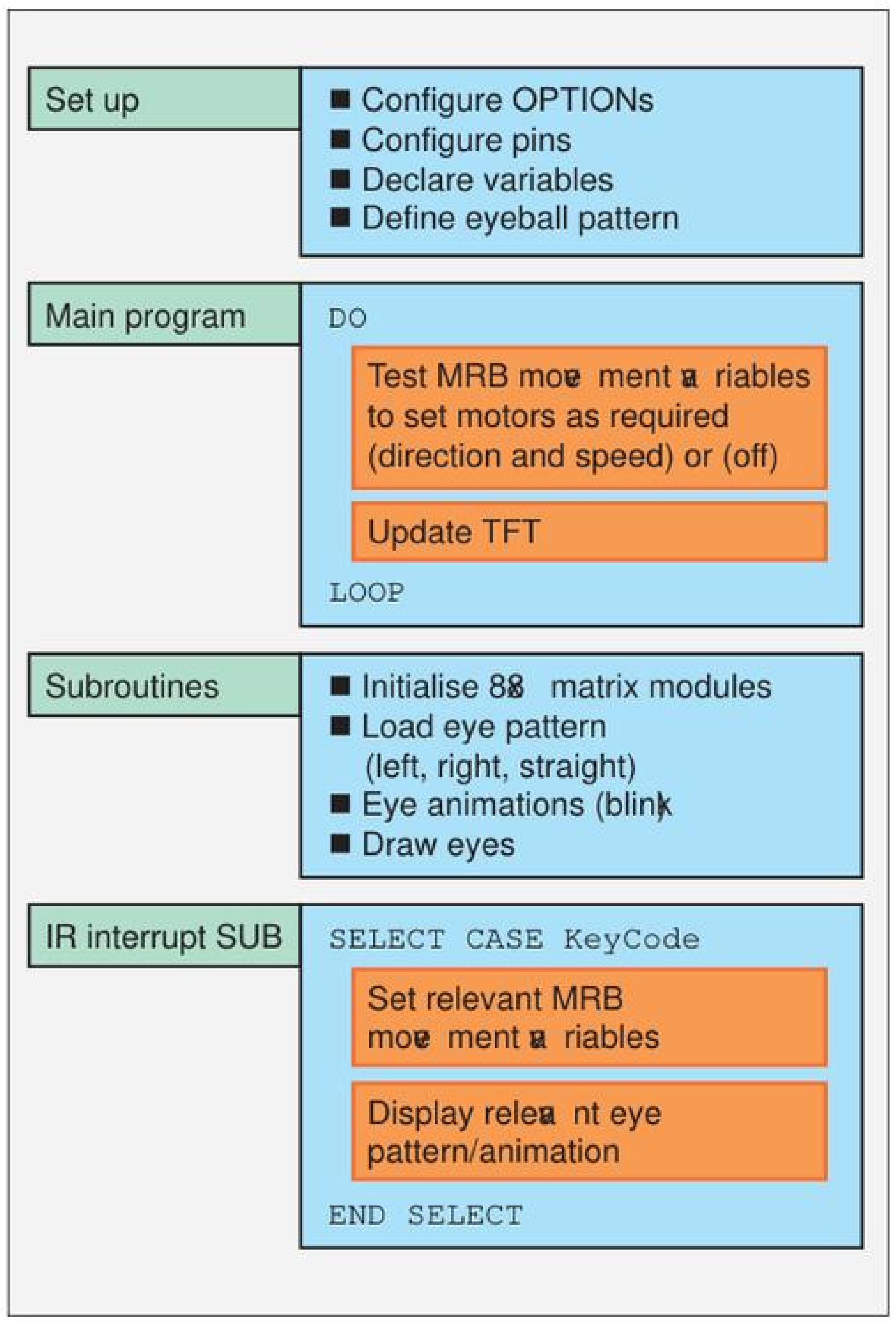

Set up
■ Configure OPTIONs
■ Configure pins
■ Declare variables
■ Define eyeball pattern
Main program
DO
Test MRB movement variables to set motors as required (direction and speed) or (off)
Update TFT
LOOP
Subroutines
■ Initialise 8×8 matrix modules
■ Load eye pattern (left, right, straight)
■ Eye animations (blink
■ Draw eyes
IR interrupt SUB
SELECT CASE KeyCode
Set relevant MRB movement variables
Display relevant eye pattern/animation
END SELECT

Fig.38. The IR controller program used this month (**MRB_IR_Control.txt**) comprises four simple sections. Do read the comments contained in the code to understand things better.

of code. The first uses some variables that define what the motors are doing:

- `MRB_Move` defines whether the robot is moving or not (`0`=no, `1`=yes).
- `MRB_Dir` defines the direction (`-1`=backwards, `1`=forwards).
- `MRB_Turn` defines which way it is turning (`-1`=left, `0`=not turning, `1`=right).
- `MRB_TurnDuration` defines how long to turn for before stopping (there are two turn buttons for each direction; one does a large step, the other a smaller step (to fine tune).
- `MRB_Speed` defines the speed of the robot. This is a PWM percentage and varies between 60% (slow) and 100% (fast) in steps of 5%. The speed is displayed on the TFT screen.

There is a series of `IF…THEN…ELSE…END IF` statements that use all of the above variables to control the four I/O pins connected to the two motors. In essence, the end result is that the robot moves, turns or stops as required.

The other block of code in the Main program puts relevant information onto the TFT. Basically, it displays the values of the above variables, converting some of them into a more 'human-friendly' format (for example `MRB_Turn` is displayed as the word 'Left' rather than the value `-1`). The presence of the `DO…LOOP` in the code means the Main program is continually using the values of all the variables. Any change in value in a variable will result in a near-instant change in what the robot is doing.

The Subroutines are used to make the Main program easier to follow. If you take a look at each of them in the program, they are all commented to guide you through what they do. The most complex one is `DrawEye` so do not panic if you don't follow the logic. It comprises a lot of logical operations on the variables discussed above. The point is, you know how to load the variables, and what effect each one has on the displayed output.

Last, the IR interrupt subroutine alters the function the robot is performing. By simply changing the appropriate variable values in the code for each `CASE`, the change in value of any variable(s) will be picked up in the Main program, resulting in the appropriate action. For example, a button can be made to act as an immediate brake (if the robot is about to crash!) by simply setting `MRB_Move=0` and `MRB_Turn=0`. Another example is a specific button that could call a subroutine that runs a sequence of frames to provide a blinking effect (try it by pressing the 'Flash' button).

That is all the detail we are going to discuss here. Have a play and see which buttons respond on the remote. Comments are in the code to guide you.

Next month

To enable our *MRB* to roam independently (without crashing into anything) it will need to have some form of collision detection. Next month we will show how easy it is to implement this feature by adding a low-cost ultrasonic distance sensor to the front of your *MRB*.

Questions? Please email Phil at:
contactus@micromite.org

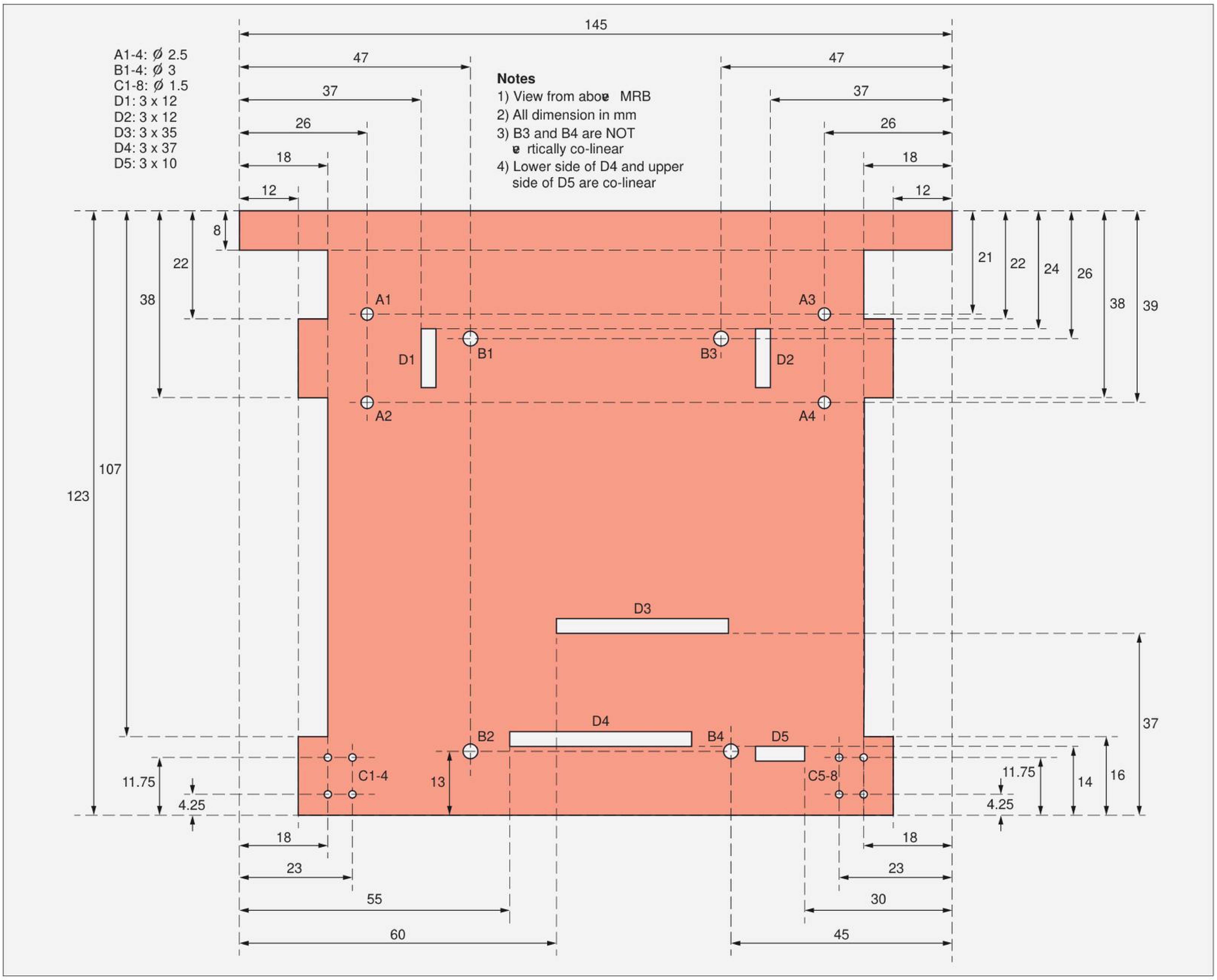


Fig.31. The *MRB* chassis dimensions – a PDF of this diagram is available for download from the August 2020 page of the *PE* website.

Practically Speaking

Hands-on techniques for turning ideas into projects – by Mike Hibbett

Introduction to surface-mount technology – Part 3

In the June issue we looked at how to choose SMD (surface-mount device) components for projects. We started with the simplest devices; passives such as resistors and capacitors. In this concluding article we will move onto the more complex task of IC and transistor selection.

First, let us dispel a common myth. SMDs are not 'different' to through-hole devices; in most cases the electronics inside the device, the sliver of silicon on which the circuit is built up, is identical. The difference is simply the package they come in, and the wires that connect the sliver of Silicon to the outside world. You might ask, 'Why have different package sizes?' There are a number of reasons.

Packaging

Wire-ended components have been a popular choice for engineers for decades because they are a logical improvement from earlier manufacturing technologies based around point-to-point wiring solutions from the valve era (see *Part 1*). As the working voltages of circuits lowered from hundreds of volts (using valves) to the 12V or lower of modern transistor circuits, the size of the circuits inside components could be reduced; one reason being that the circuits did not require large track widths and spacings to avoid voltage breakdown. Just as improvements in semiconductor technology brought lower operating voltages, they also reduced current consumption, at least for digital devices. A few exotic semiconductor devices still switch high voltages and high currents, but they are rare, and it would be unwise to work with them unless you really know what you are doing!

The lowering of device voltage and current consumption also lowered the diameter of the wires connecting the surface of the silicon to the pins on the device package. Since the circuits to which these devices are connected to have thin tracks, then so too can the packages in which the components are placed. Fig.1 shows some examples – a BC547 in the familiar TO92 package and also in the SMD SOT23 package. Same device, different wires attached to it in the packaging. As you can see in Fig.2, the two package images in Fig.1 are not shown to the same scale! Both of these transistors are still in high-volume production, and both are in high demand.

So why has the much smaller and cheaper SMD variant not rendered the older through-hole part obsolete, like valves? Through-hole components still have value in the global electronics industry. Old product designs that use through-hole components are still working, still valuable, so why change? Examples include railway station ticketing machines, electronic access gates and fire alarm systems, to name a few. As the saying goes, 'if it ain't broke, don't fit it!' and there are many electronic systems in use globally that were designed 30 years ago, and which are still commercially valuable.

There are some specific cases where through-hole components still dominate in modern designs. The main example is in high-energy systems, where transistors switching very large currents, such as heating controllers, require thick wires to pass the large currents. For the majority of hobbyist projects, however, very large current handling is rarely a design requirement.

The demand for SMD

The move to SMD components was driven by the consumer electronics market, specifically audio, computing and communication devices, where very large currents are not a factor (excluding power amplifier circuits, of course) and the sheer quantity of potential sales drove research, both academically and within industry, to improve the efficiency of electronics manufacturing, where efficiency measures not only cost, but also reliability, size, weight, and the ease and speed of manufacture.

That research and innovation drove the need to be able to place components onto PCBs quickly. Machines already existed that could place wire-ended components onto a PCB, cut the leads back and pass them through a soldering bath. The technology developed to place wire-ended components into PCBs is incredible. Check out this YouTube video to see it in practice: **https://youtu.be/ZFy0b-Jw4Ec**

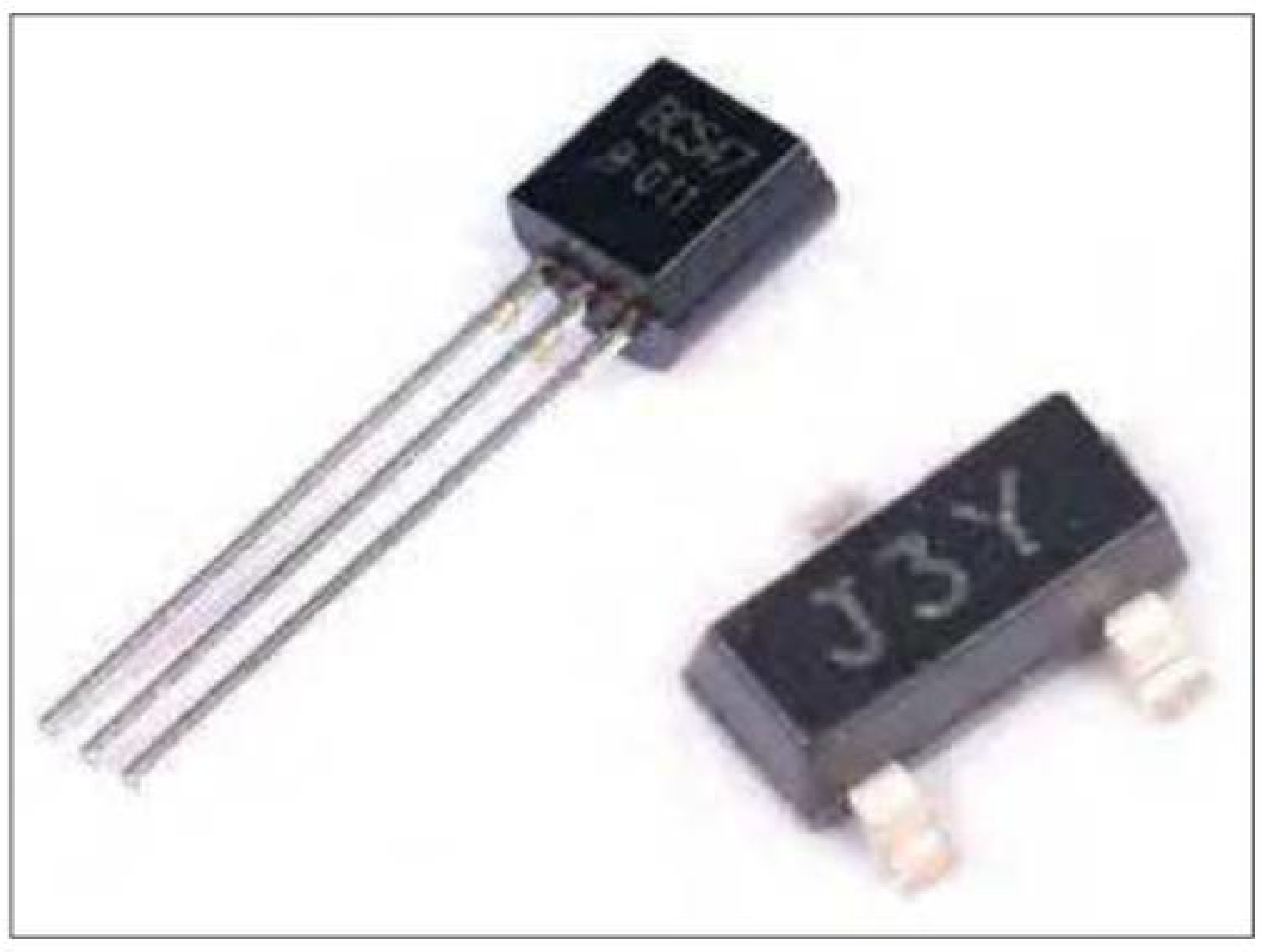


Fig.1. Two variants of the same BC547 transistor, but at very different scales.

Although clever through-hole systems are relatively slow and error prone, and with demand for the production of electronics devices increasing, the ability to move to component packages which could be machine placed more easily became a world-wide objective. International standards and cooperation brought us quickly to where we are now, with new package types – without wires – and new production machinery to handle them.

On a personal level I feel it was the combination of international cooperation coupled with consumer demand that made this move to SMD production possible, in such a short timescale. Just a few decades.

The general trend in IC packages can be seen in Fig.3. The devices in the bottom

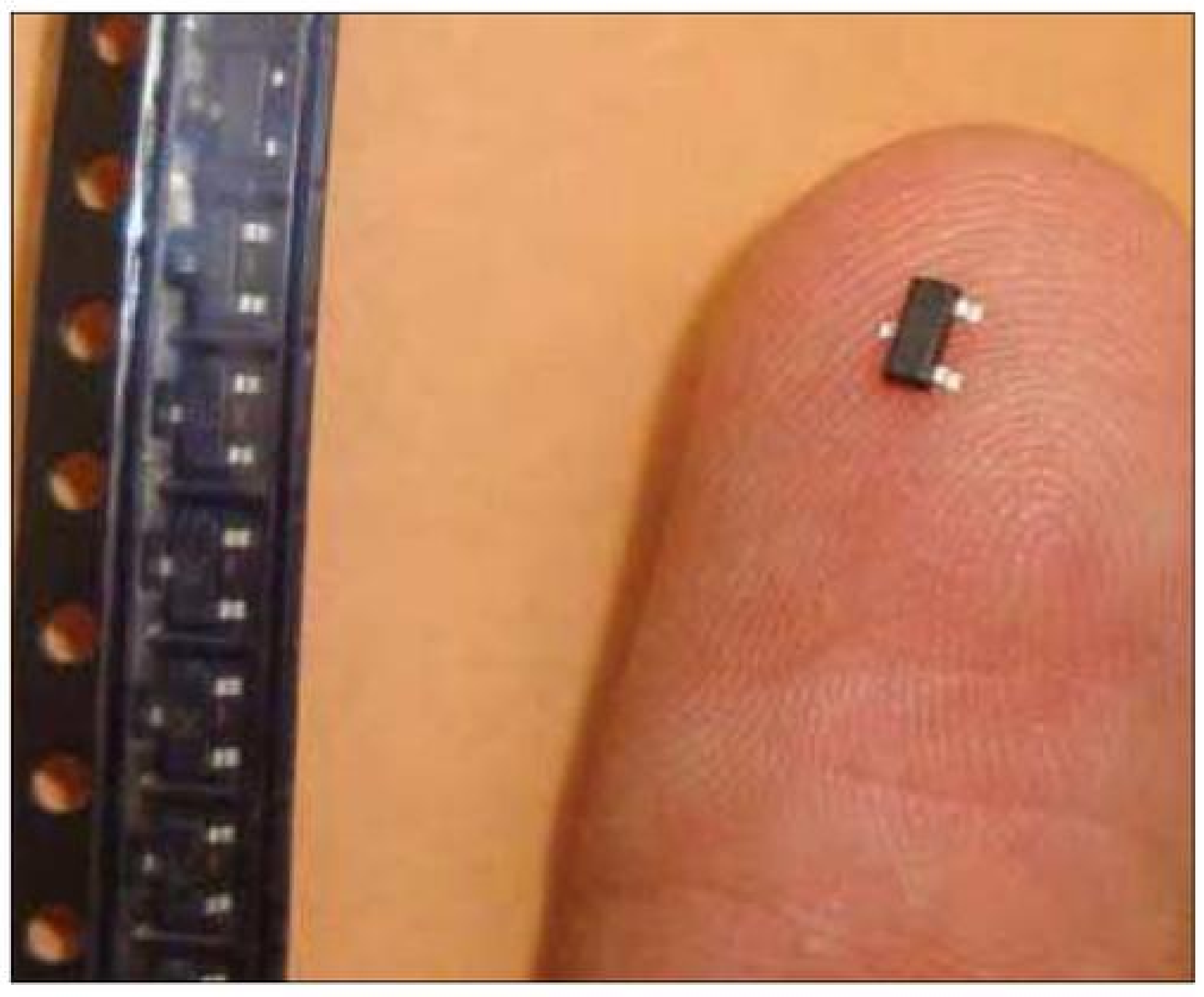


Fig.2. SMD transistor, to scale.

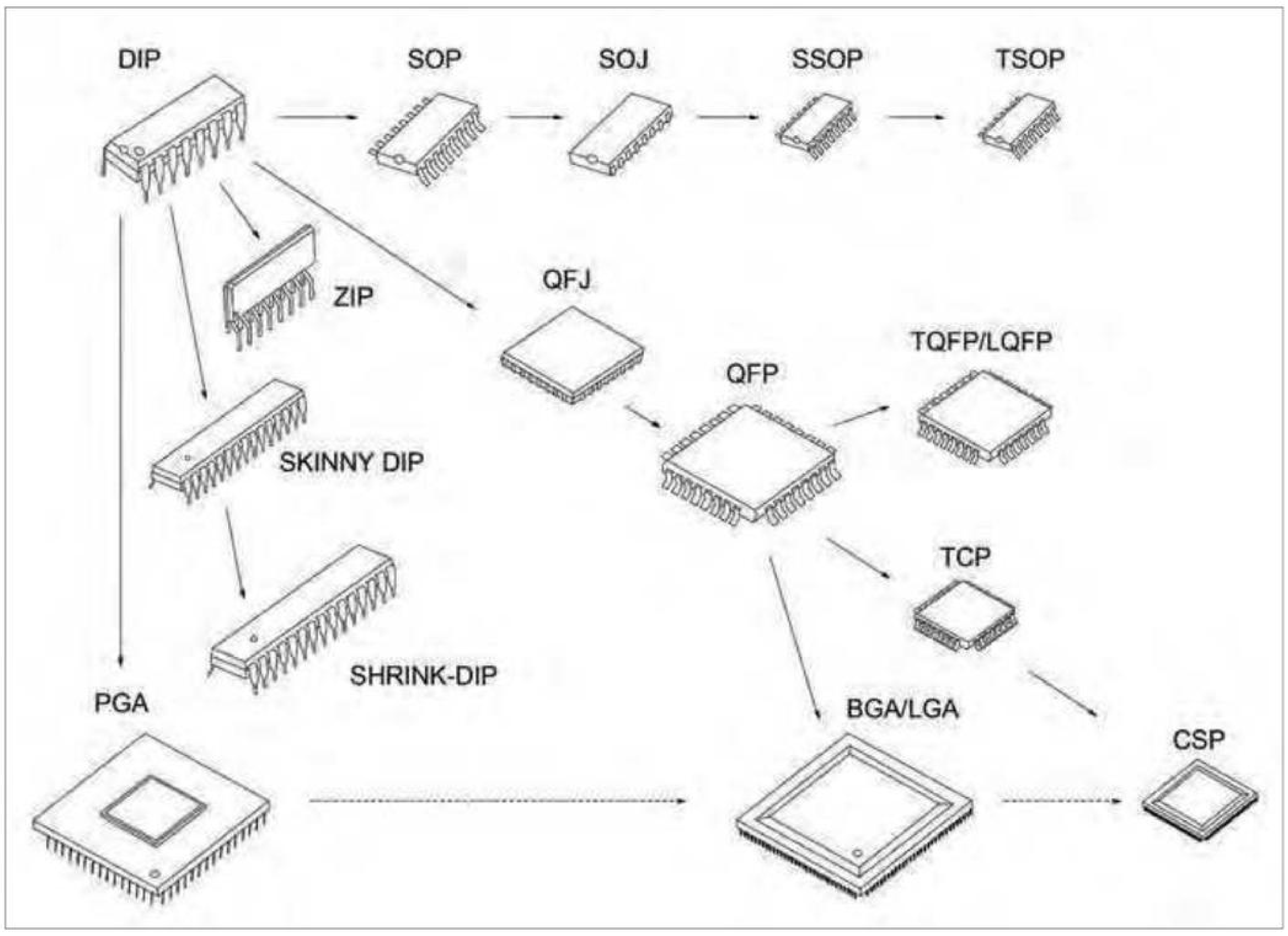


Fig.3. IC package trends from classic DIP to assorted SMD varieties (image: Microchip).

row are beyond normal hobbyist application because they demand expensive 4-layer or more PCBs, expensive CAD design tools, and a lot of experience. These tend to be highly specialised devices, so we are not missing much by avoiding them.

While new through-hole devices have been supplemented by surface-mount devices and continue to be available in both package types, some new devices are only available in SMD. This is because surface mounting does provide some unique benefits. Shorter wire lengths means devices can run at higher frequencies, which can be beneficial for high-frequency radio parts, or fast microcontrollers. Some devices are designed for specific uses – like smart-phones or personal computers – so it makes no sense to offer the parts in older package formats. On the other hand, for a voltage regulator it may make sense to offer the part in both case styles, but for something like a camera chip or a Flash memory device, the target applications do not include old-style manufacturing techniques.

So, as interesting ICs come onto the market, our choice as hobbyists is to either learn new soldering techniques or rely on the likes of Adafruit or Sparkfun to produce a small, simple 'plug-in' PCB, holding nothing other than the IC and its necessary components. If you are not tight for space in your project design, this plug-in PCB approach is perfectly reasonable – and there are many far-east suppliers making these 'break-out' boards available on eBay for this exact purpose. Fig.4 shows a typical board available from Sparkfun. Just the IC and two capacitors, but with a 0.1-inch SIL header strip allowing it to be placed onto a regular breadboard. Two mounting holes enable it to be fixed within a project enclosure.

Whatever the package, some things never change. Whether you are buying a chip or a PCB, you will still face the same issue – is this device right for your design, and will it work with your other components easily? To understand this, you are going to have to delve into the manufacturer's datasheet.

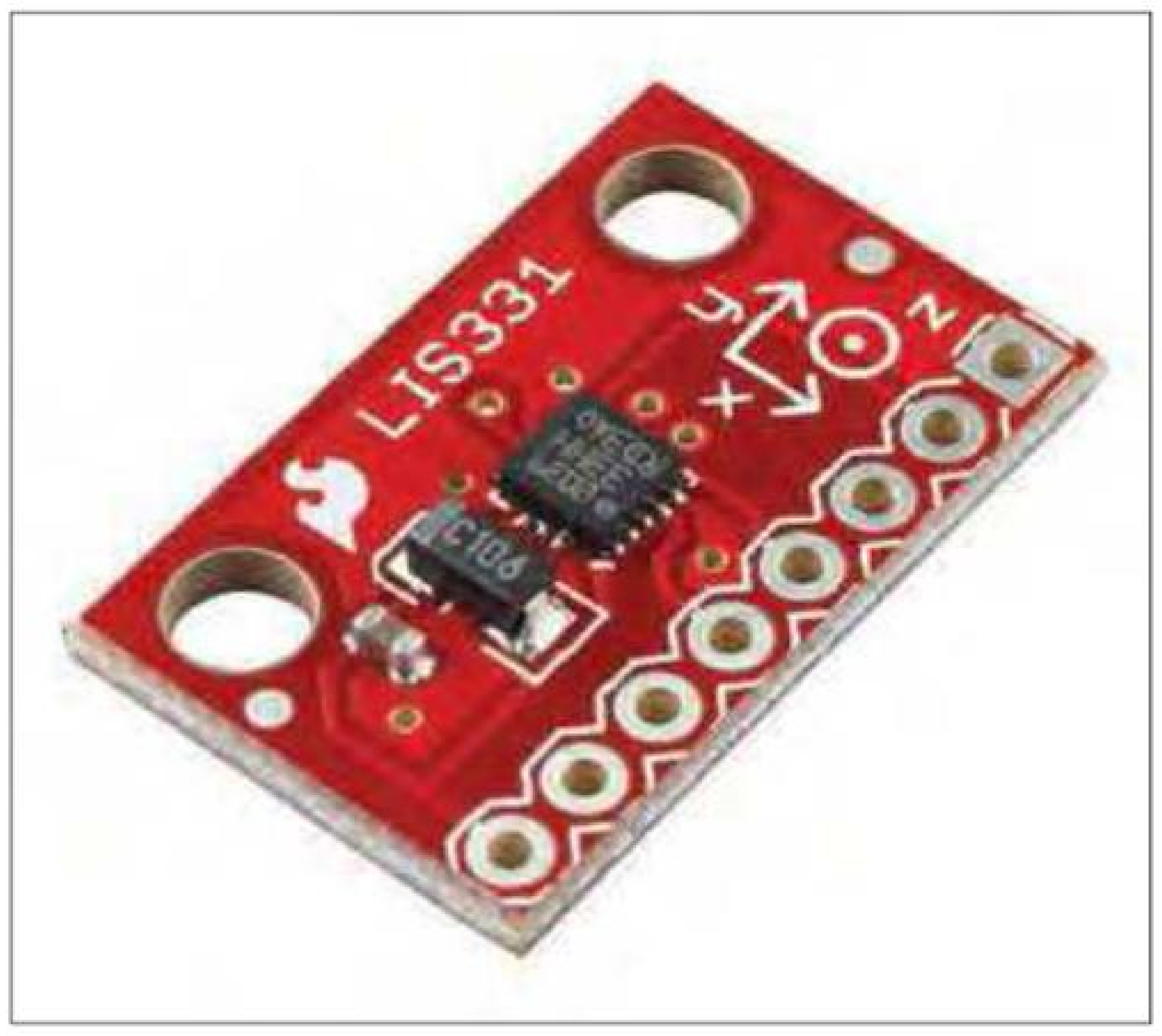


Fig.4. SMD breakout boards are the hobbyists new friend!.

Datasheets

Datasheets follow a fairly standard format, containing similar information even across different types of device technologies. This information can be summarised as:

- Functional overview
- Device variants listing
- Absolute maximum values
- Detail of operation
- Reference application circuit
- AC/DC characteristics
- Temperature – always has an impact
- Device packaging specifications
- Shipping packaging specifications

Some of these sections are obvious. Let's dive into the sections that are less so.

Device variants

This section details the different manufactured variants of the device that are available. For microcontrollers this can be memory size, I/O pin count, and even the maximum operational frequency. It's not uncommon for some of the features listed in the datasheet to be missing on smaller pin count package variants, so take care when choosing your package. Each device variant will have a unique order code, which will be listed in the datasheet.

Absolute Maximum values

These parameters list the voltages, frequencies and temperature range outside of which the device is almost certain to be destroyed, or not function. A 5V microcontroller may have an absolute maximum working voltage of 6.5V. *Never* design your device to the values in this section of the datasheet.

AC/DC characteristics

In this section the manufacturer provides examples of what the operational parameters will be like when the device is in use. This will include current consumption and maximum operating speed, usually given at different voltages and temperatures. This section is critical for understanding, for example, what size of battery or power supply you will need to power your circuit.

Functional overview

This is the meat of the datasheet; it's the section you need to be fully familiar with before designing the part into your circuit. How long this section is will depend on the complexity of the part. We've listed a cross section of device datasheets at the end of this article, ranging from a seven-page transistor datasheet, through a 40-page LM358 op amp datasheet up to a 600-page (!) microcontroller specification.

20 years ago, you would have needed to be intimately familiar with every page of that 600-page document, but in recent times the industry has moved to making the designer's job simpler. Datasheets typically provide an example application circuit, which invariably becomes the starting point for designers. For simple circuits this often forms the complete electrical design for your project. It's also not uncommon for a recommended PCB layout to be provided, especially for power, audio or RF ICs. When a recommended PCB layout is provided always follow it as closely as possible; in our experience the term 'mandatory' would have been better than 'recommended'.

More complicated ICs will have separate application notes that provide the design of complete reference products, completely free to copy into professional

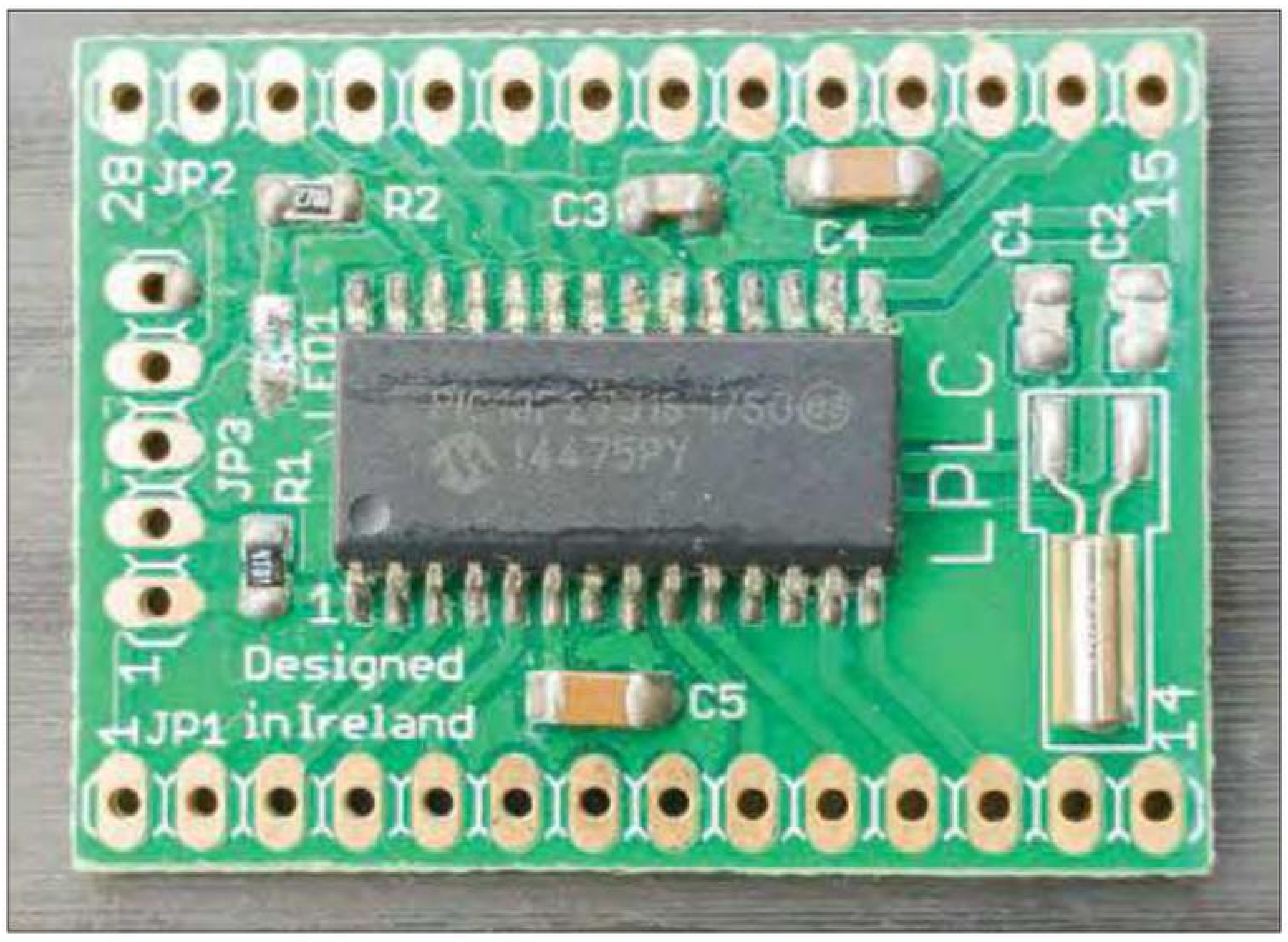


Fig.5. Manually placed SMD components and manual soldering.

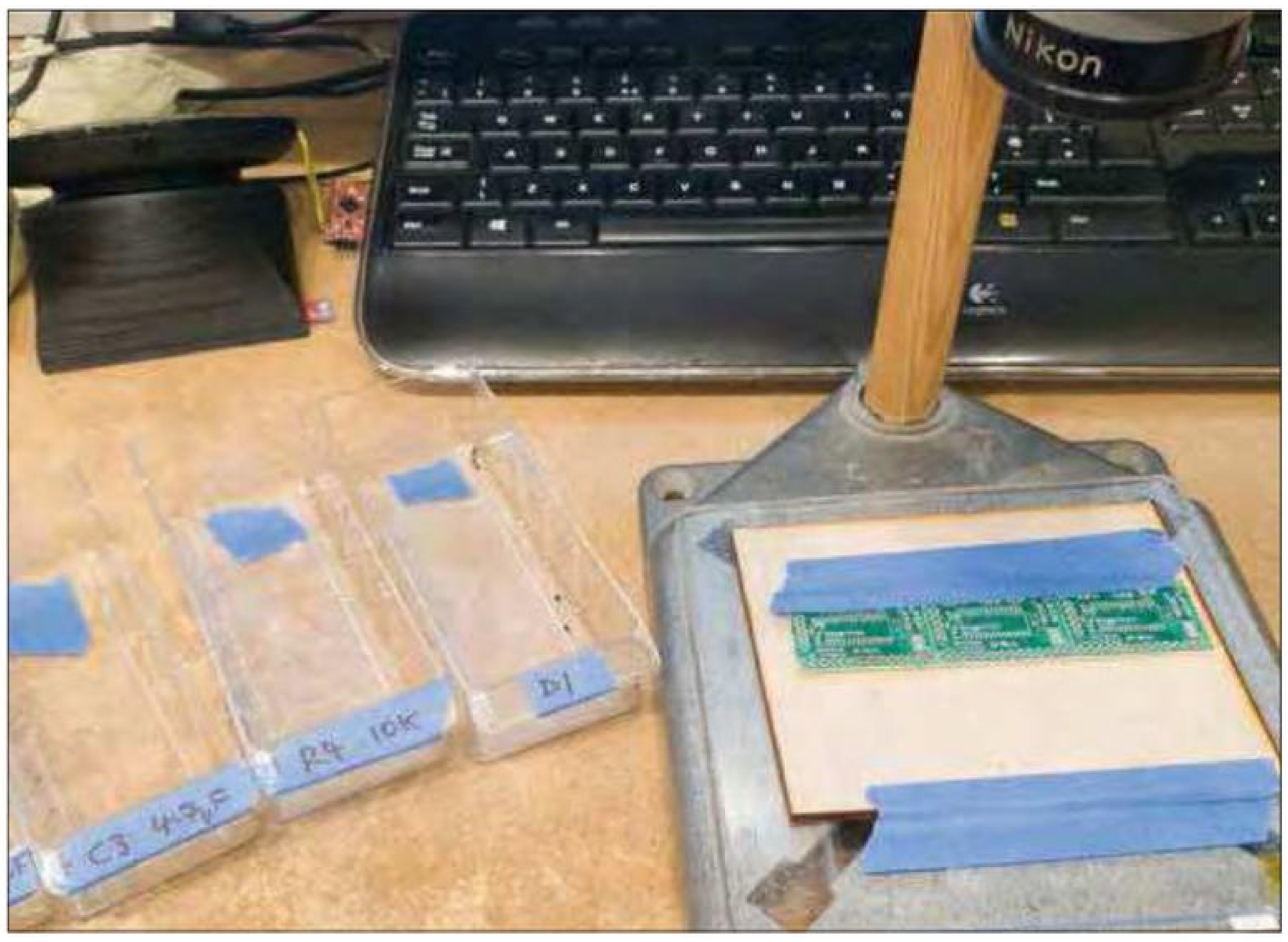


Fig.6. Organisation is crucial when assembling PCBs with SMD parts.

designs. It is your responsibility to ensure the circuit works in your actual application, but often 90% of the design work is done for you, including source code for the software to make use of the IC's complex peripherals, again with example product designs, free to use.

A final point on datasheets: some components are 'generic', meaning they are produced by different manufacturers, so be careful to read the actual manufacturer's datasheet for the part you are using. Voltage regulators, 555 timers and 358 op amps are just a few examples of devices supplied by different manufactures. Subtle differences in, for example, peak working voltage or quiescent current consumption, are normal and can impact your design, so remember to look at the *correct* manufacturer's datasheet.

PCB assembly with SMDs

In general, when creating a PCB with SMD parts you will be using a CAD program. For example, you might use popular (and free) applications such as EagleCAD or KiCAD to design your board. These tools come with standard component footprints, but do be aware that there are a number of different pad sizes available for a given component size, such as 0805 resistors and capacitors. There are ISO standards that define large, medium and small footprints – relative terms, of course! A component supplier will always have their own specific design recommendations within their datasheet.

For high-component-density designs, and to minimise the risk of soldering failures, a designer will normally follow the component supplier's recommendations. They will discuss the proposed design layout with the PCB assembly company they intend to use, to ensure the design makes best use of the machines that will be used to solder the parts. As hobbyists, we are not required to focus on the high-volume manufacturability of our designs; we want to make them as easy to hand solder as possible. With this in mind, look to use a component footprint that leaves exposed copper on the pad for easy access with a soldering iron. Fig.5 shows a board designed by the author that was hand assembled. Notice the large amount of solder on the pads of C5; this part took just a few seconds to hand solder.

Fig.5 highlights another important point with board designs using SMD component placement. The capacitors surrounding the IC are close to it, as they are required to be. They could have been a lot closer, but if they were, it would have made soldering to the IC pins difficult without causing shorts. Even with this design the IC had to be hand placed first, so that the capacitors did not restrict access to it with a soldering iron.

Organising your components during assembly is critical. Only resistors come with a clear value indicated, capacitors in the main do not. In Fig.6 you can see the author preparing to assemble several identical PCBs. Parts are assembled in 'batches' of components to minimise the clutter on the workbench; each component placed in a container, clearly marked. Tweezers are used to pick a component from each container, placed onto the PCB and soldered. If a component is dropped, don't waste time looking for it – pick another!

With surface-mount work, where you need a visual aid to do the work we find the best process is to lay the PCB flat under a microscope or magnifying glass, and keep the PCB fixed in place with masking tape or blue-tack. When assembling several boards of the same design we typically resort to building a simple jig out of balsa wood. The jig is stuck in place and the PCB dropped into the recess. This works very well and is cheap and easy to do. The walls of the jig are glued in place with super-glue, so a jig can be manufactured in minutes.

Advanced selection

As you move from hobbyist projects into professional circuit design and product creation, additional requirements come into play. It's important to understand if the part you are selecting meets the standards of the countries you are going to be selling in – for example, is the device lead free (ROHS compliant), is it available in reels, and if so, will those reels fit on your manufacturer's machines? All of this information is available in the datasheet or documents the datasheet refers too.

Reels of components are always supplied in airtight wrappers. If an SMD component is exposed to air, moisture can creep into the package and potentially destroy the device when rapidly heated in a production SMD oven. The degree of sensitivity to moisture ingression is related to the device package design and is always stated on the datasheet. As a rule of thumb, we never open sealed component packs destined for our manufacturer!

In summary

The industry trend is to continue to make small, close-pin pitched ICs that will continue to pose challenges for hobbyists. Thankfully, the global hobbyist market is large enough that it is commercially viable for electronics manufacturers to supply break-out boards for a wide variety of exotic ICs, so we hobbyists will not be excluded from taking advantage of the latest advances in IC technologies. You can expect to see more use of break-out boards in *Practical Electronics* articles, including the current *Pic n' Mix* series.

References to datasheets

https://infocenter.nordicsemi.com/pdf/nRF52833_PS_v1.2.pdf
https://www.mouser.com/datasheet/2/149/BC547-190204.pdf
http://ww1.microchip.com/downloads/en/devicedoc/22008e.pdf
https://www.ti.com/lit/ds/symlink/lm358-n.pdf

Max's Cool Beans

By Max the Magnificent

Flashing LEDs and drooling engineers – Part 6

Well hello there! How nice it is to see you again. May I make so bold as to say that, as a member of the *Cool Beans* community, you manage to look both awesome and highly intelligent. This isn't a look many people can carry off successfully, so kudos to you!

Out of date

As I mentioned in my previous column (*PE*, July 2020), my current hobby project is to build a 12 × 12 = 144 array of ping-pong balls, each containing a tricoloured LED in the form of a WS2818 (aka 'NeoPixel'). As you may recall, I also introduced the Seeeduino XIAO microcontroller with which I plan to drive my array. The XIAO is only about the size of a standard postage stamp, but it boasts a 32-bit Arm Cortex-M0+ running at 48MHz with 256KB of Flash (program) memory and 32KB of SRAM. Yummy scrummy! In fact, I'm so enthused by this little beauty that I was moved to create a video (**https://bit.ly/373vPQC**).

One slight complication is that the XIAO's input/outputs (I/Os) are 3.3V, but I need a 5V signal to drive my pixels. Happily, I found a rather cool hack that allows me to use a 1N4001 diode and a 'sacrificial' pixel to implement a cheap-and-cheerful 3.3V-to-5V voltage-level converter (**https://bit.ly/3cxMhcV**).

In anticipation of finishing my ping-pong ball build (and that's not something you expect to hear yourself saying very often), I bounced over to the XIAO's wiki webpage (**https://bit.ly/30bCyGH**) to learn how to add this little scamp to my Arduino IDE. Next, I created a really simple test sketch (program) to tickle my pixels, selected the XIAO as my target board, and hit the 'Compile' icon.

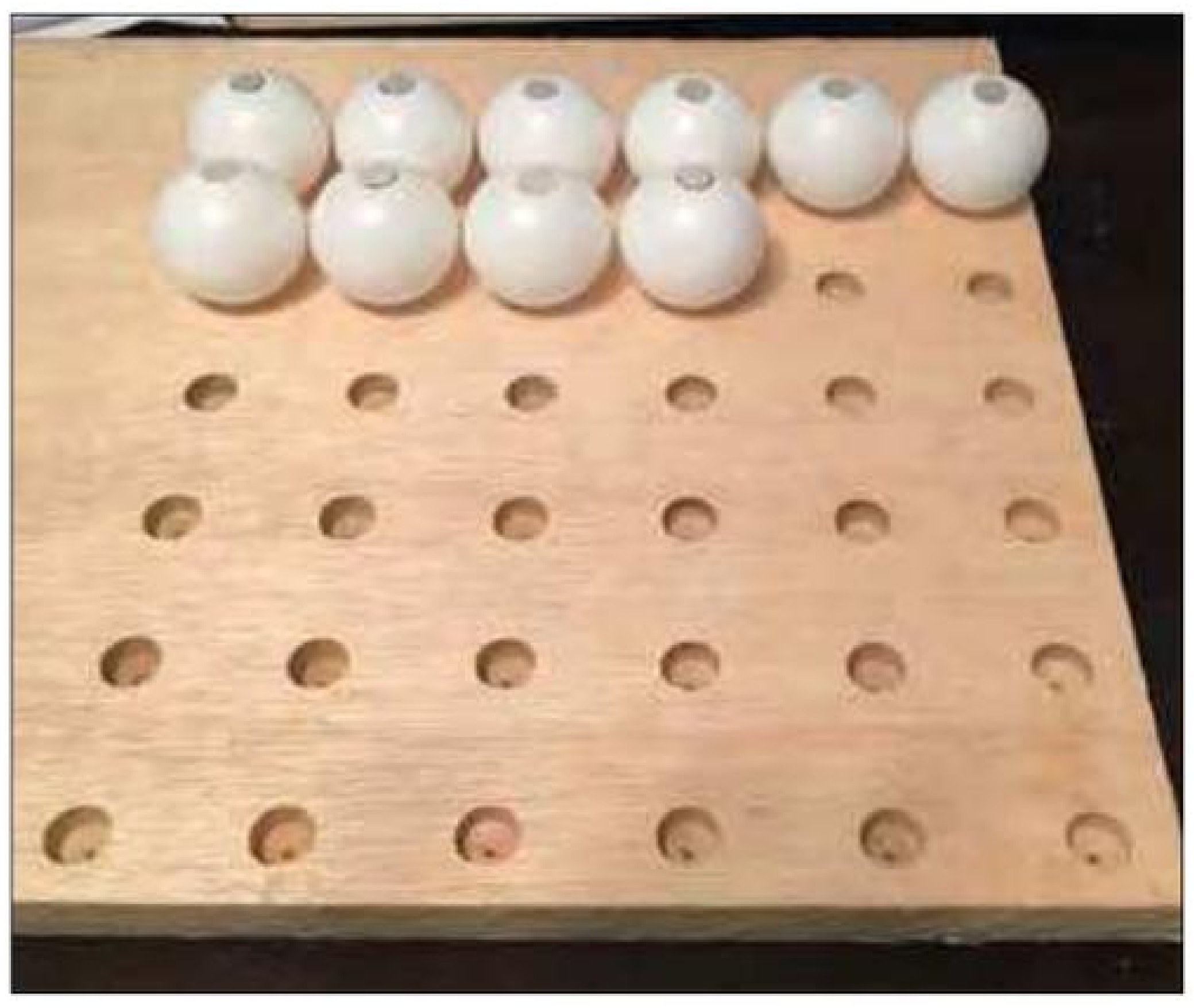


Fig.1. A jig to hold the ping-pong balls.

Sad to relate, the compilation failed with multiple errors and grim warnings of a type I've never seen before. 'Oh dear,' I said to myself (or words to that effect). I switched the target board to an Arduino Uno and the compilation passed as expected. It also passed when I switched back to the XIAO and commented out any NeoPixel-related statements. Hmmm.

By this time, I was wearing my sad face, so I emailed my sketch to the folks at Seeed Studio explaining my conundrum. The very next morning, I found an email in my InBox from field applications engineer (FAE) Anson He, who said that my test program had compiled for him with no problem. Anson recommended that I check to see if I had the latest version of the Arduino IDE (I had) and the latest version of Adafruit's NeoPixel library (I hadn't).

After removing my old NeoPixel library, I followed the instructions on *Adafruit's NeoPixel Überguide* (**https://bit.ly/2XzuqOB**) to install the latest and greatest version of their library. Following this, everything compiled like a charm, at which point I dispatched the butler to fetch my happy trousers (I can't perform my Happy Dance without them). Now, all that remained was to actually construct the array, which sounds easy if you say it fast and gesticulate furiously.

Hot glue is my friend

While my software gremlins were underway, the hardware construction proceeded apace. Unfortunately, I hadn't given as much thought as perhaps I should to how I was going to precisely position the ping-pong balls. As you may recall from my previous column, rather than drilling the holes in the balls to accommodate the NeoPixels, I cut them by hand using some small curved nail scissors. The thing is, that this was easier to do prior to mounting the balls on the board, but it left me with the problem of aligning everything.

Prior to this, I had no idea how tricky it can be to corral 144 ping-pong balls and bend them to your will. The little rascals seem determined to escape. After experimenting unsuccessfully with a variety of different schemes, I created a jig by drilling a 6 × 6 matrix of smaller holes in a piece of scrap board and using this to position and restrain the balls (Fig.1).

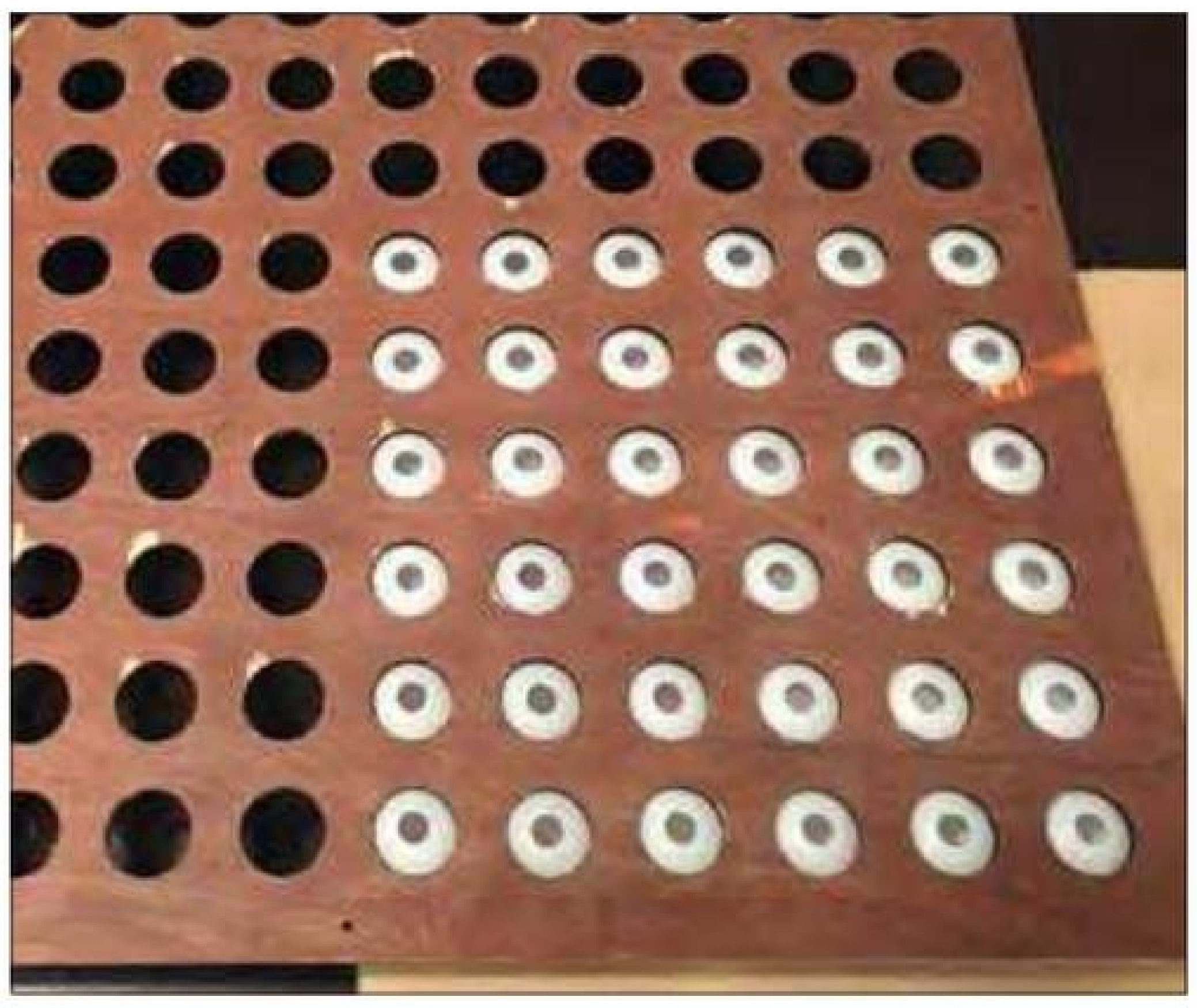


Fig.2. Ready to attach the first 36 balls.

Next, I laid one corner of the main board on top (Fig.2), used a bit of wooden dowel to tweak the position of the holes in the ping pong balls so they pointed straight up, and used my hot-glue gun to fix everything in place. I then repeated the process for the remaining three quadrants.

As an aside, I'm regarding this 12 × 12 array as being a prototype to 'iron out the wrinkles' in the process. In the fullness of time, I'm hoping to build a wall-size display. One thing I've decided is that the next time I do this, I will create smaller 8 × 8 panels and then join them together, in which case the alignment jig will be the same size as the front panel.

The boring bits

Every project involves one or more boring bits, and this is where I now found myself.

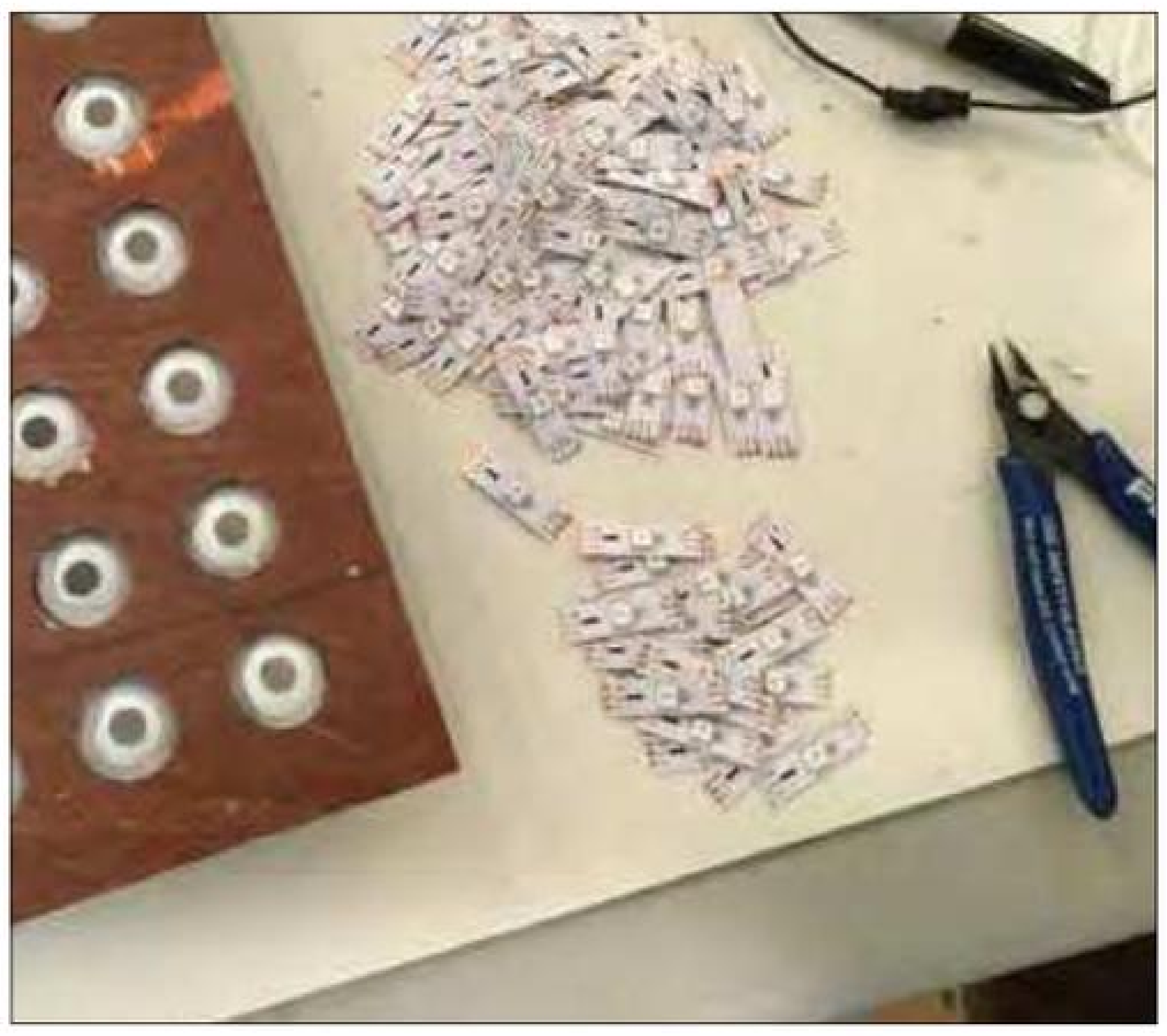


Fig.3. Cutting the NeoPixel Segments.

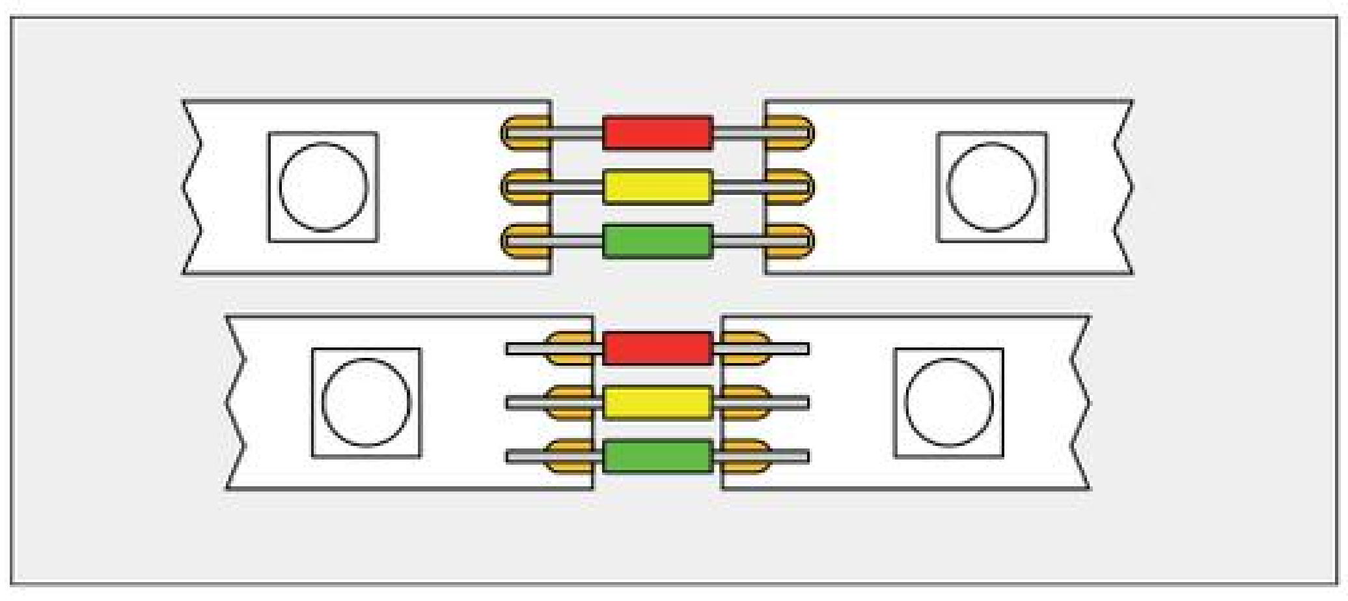


Fig.4. Creating linking wires to accommodate different gap sizes.

It started with snipping out 145 segments from my NeoPixel strip – 144 for the array, and the ‘sacrificial’ pixel for the voltage-level converter (Fig.3). This is less fun than you might expect, because the strip is delivered wrapped in a protective plastic cover making the pixels waterproof so they can be deployed outside, and this cover has to be painstakingly removed (not that I’m complaining, but it all takes time).

The next step was to use more hot glue to attach these segments to the ping-pong balls. Remembering that the pixels are going to be daisy-chained together, the way I did this is to align alternate rows in different directions, first right-to-left, then left-to-right, then right-to-left, and so forth. This keeps the signal wires short when connecting the end of one row to the start of the next.

In the case of the body of the array, I needed three short wires to connect adjacent pixels: 5V, 0V, and the data signal. The problem here was that there are variations caused by things like the segments being cut to slightly different lengths and the holes in the ping pong balls not being precisely aligned. As a result, some gaps were smaller, while other gaps were larger. Rather than custom-create each piece, I decided to make the wire parts long enough to span the widest gaps, while the insulation parts were short enough to fit in the narrowest gaps (Fig.4).

So, excluding the ends of the rows, we have to make 11 groups of 3 connections on 12 rows, which equals creating 396 little wires (there’s a couple of hours I’m not going to see again). My reasoning for leaving a small piece of insulation was that, while attaching these wires, I didn’t want my long-nosed pliers to act as heat sinks resulting in bad joints. In hindsight (the one exact science), I discovered this really wasn’t an issue, and I could have saved myself a lot of effort by simply using short lengths of uninsulated tinned copper wire cut from a roll. Be this as it may, I created videos showing the process of creating the wire connectors (**https://bit.ly/2MCQ7al**) and attaching them to the array (**https://bit.ly/3dHzgPt**).

Feel the power!

This is where things started to get interesting again, because I was now on the home stretch of connecting the power.

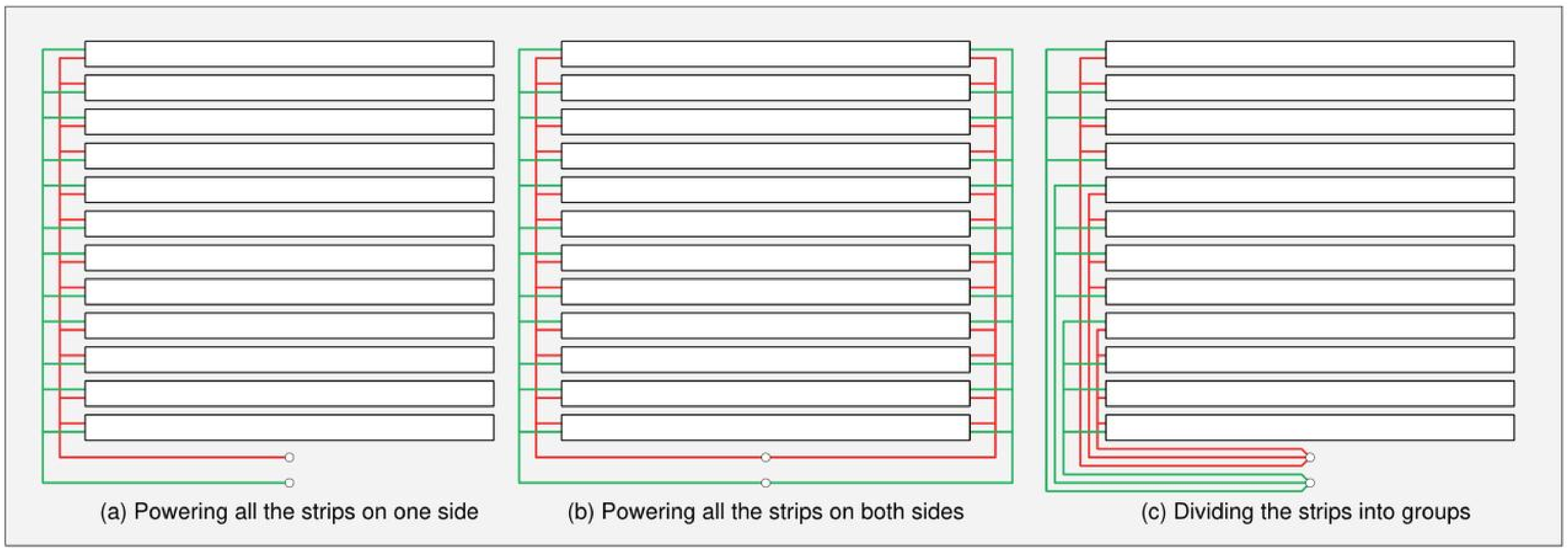
Each NeoPixel contains red, green, and blue sub-pixels. The data sheet says that, when full on, each sub-pixel consumes 20mA, so each NeoPixel will consume 60mA. To be honest, when I’ve measured this in the real world, I’ve never seen more than 45mA for a fully on NeoPixel, so this is the value I usually go by myself (not that I’m recommending this as a practice for anyone else, you understand).

One further consideration is that, in use, it would be rare for us to have all of the elements in the array fully on. Most of the time, I would be surprised if we averaged more than 20%. Having said this, we should always design for the worst-case scenario (I know this goes against my using 45mA vs. 60mA. What can I say? We live in a crazy mixed up world).

We have 145 pixels if we include the ‘sacrificial’ pixel, even if we’re not planning on using it for anything. The worst-case full-on scenario according to the data sheet would be 145 × 60 = 8.7A. By comparison, the worst-case scenario assuming 45mA per pixel would be 145 × 45 = 6.6A.

I had originally been planning on powering all of the strips on one side as illustrated in Fig.5a (note that the reason the power and ground wires alternate top to bottom from strip to strip is that alternate strips have their signal paths going right-to-left and left-to-right, which also swaps the orientation of the 5V and 0V signals.

Unfortunately, the only solid-core copper wire that I had to hand in my treasure chest of bits and pieces, and that I’m using to supply the power, is rated at 3.5A. Of course, as with anything in electronics, there are myriad ways in which we could address this issue. One way would have been to stick with the scenario depicted in Fig.5a, but

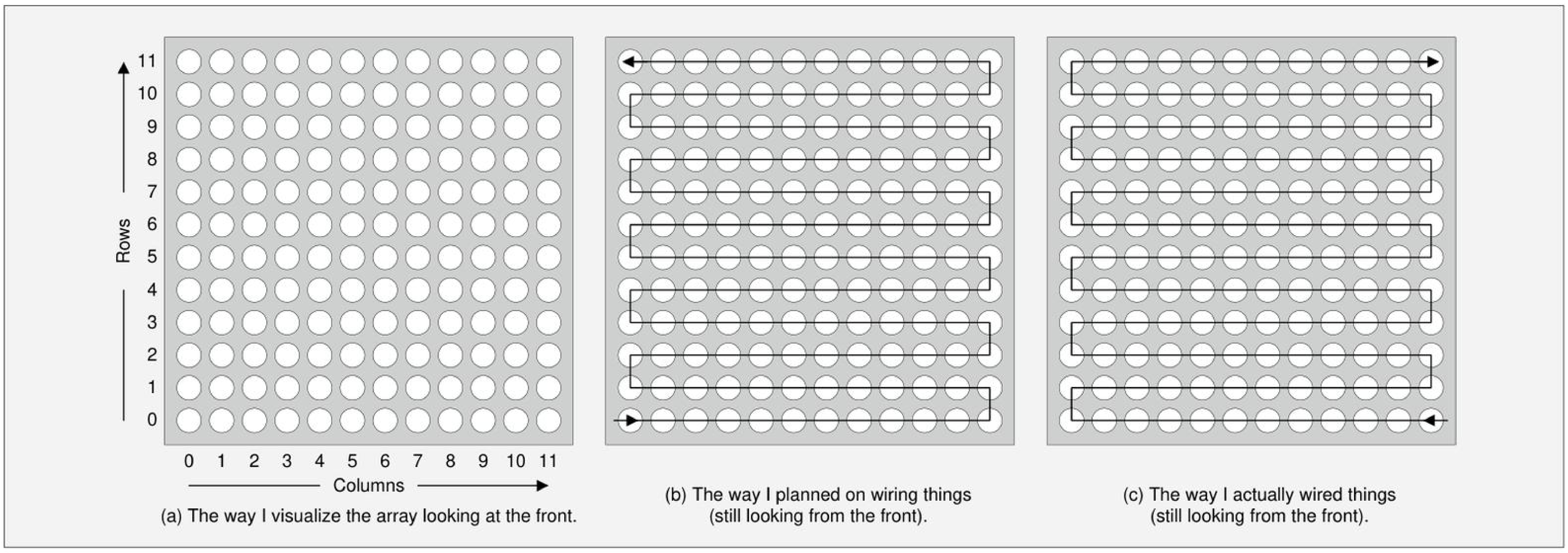


(a) Powering all the strips on one side

(b) Powering all the strips on both sides

(c) Dividing the strips into groups

Fig.5. Different power wiring scenarios.



(a) The way I visualize the array looking at the front.

(b) The way I planned on wiring things (still looking from the front).

(c) The way I actually wired things (still looking from the front).

Fig.6. When plans meet the real world.

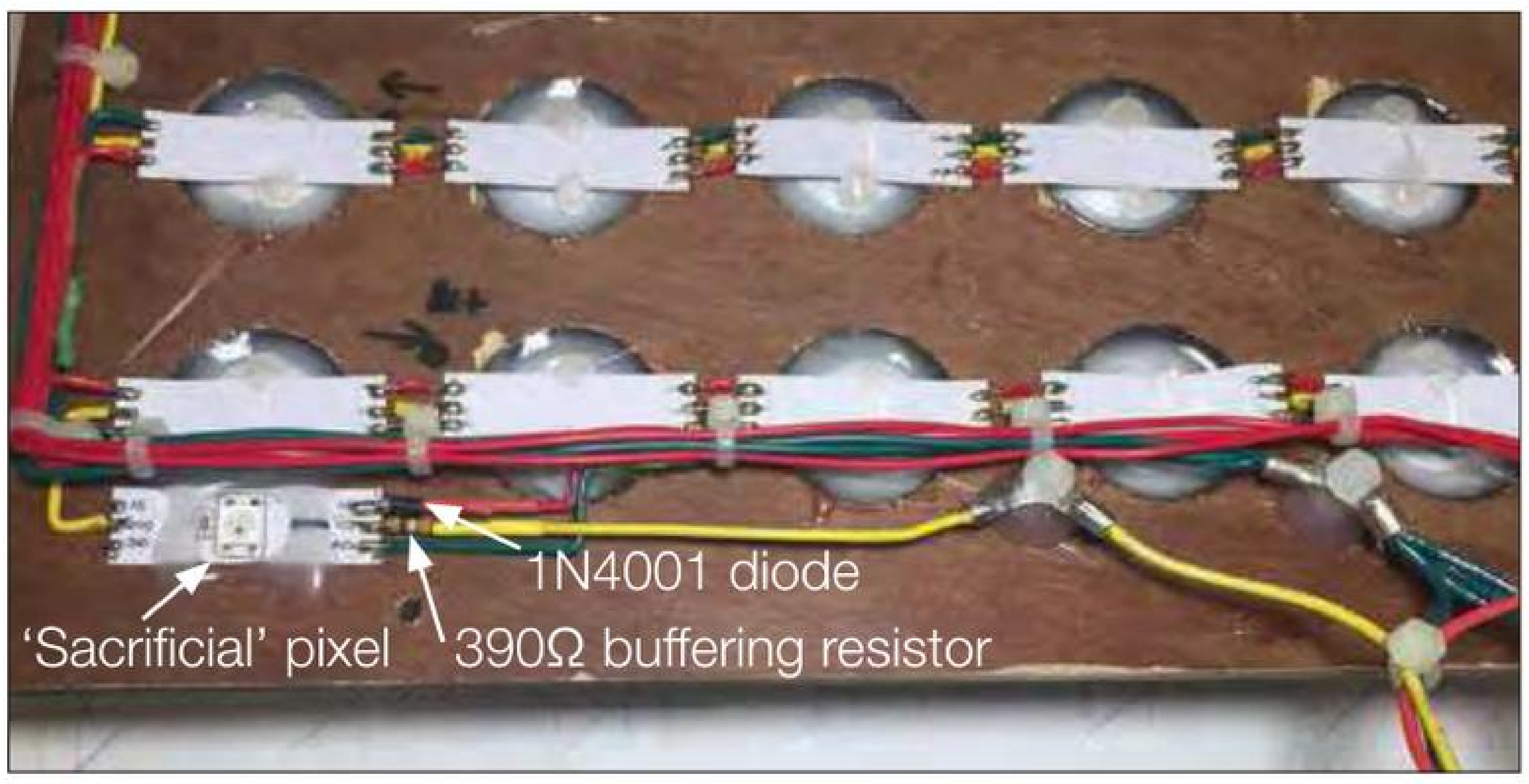


Fig.7. It's your other left!

to double up on all of the wires. This would give us a 7A capacity, which would satisfy our 6.6A requirements.

An alternative would be to stick with single wires, but to wire one set on the left of the strips and the other set on the right as illustrated in Fig.5b. This scenario, which also provides a 7A capacity, has the advantage that, if one of the power or ground connections in the middle of a strip were to fail (eg, due to a bad solder joint), then all of the pixels in that strip would still be powered.

The approach I eventually opted for, and which I documented in a video (**https://bit.ly/2Y6L9bh**), was to divide the strips into three groups of four and to power each group independently, as illustrated in Fig.5c, thereby providing me with a 10.5A capability, which more than satisfies even the worst, worst-case 8.7A requirements associated with all of the pixels being fully on while each consuming 60mA. Phew!

Your other left!

The way I've been thinking about my 12 × 12 array is as a matrix of 12 rows and 12 columns. If I'm looking at this from the front, element (0,0) will be in the bottom left-hand corner as illustrated in Fig.6a. Based on this, I was planning on wiring things such that – still looking from the front) – the first pixel in the chain was located in the bottom left-hand corner, as illustrated in Fig.6b.

Unfortunately, when I sat down at the workbench (aka the kitchen table) and set to connecting all of the wires, I forgot I was looking at the back of the array and I located the first pixel in the chain in the nearside left-hand corner (Fig.7). Observe that the lone pixel nearest to us is the 'sacrificial' pixel that's acting as the voltage-level converter. We can (just) see the black 1N4001 diode connected between the 5V supply and the power input to this pixel; also, the 390Ω resistor buffering the data signal from the XIAO microcontroller. So, as a result of all these shenanigans, the way I actually ended up wiring the array is as illustrated in Fig.6c.

Of course, none of this really matters because we're going to be finagling things in software anyway, but it still irritates me that I lost track of things in this way.

Testing, testing...

Once the wiring was completed, I was ready to perform my initial tests. It always pays to keep one's first test as simple as possible, so my equivalent of the classic 'Hello World' program was to simply light each pixel in sequence. Normally, our 144 pixels would be numbered from 0 to 143. However, since we actually have 145 pixels, with the 'sacrificial' pixel occupying location 0, the pixels in our array are numbered from 1 to 144 (Fig.8).

At the heart of the first test program is a function shown below. As we would expect, the result is to light the pixels in a serpentine pattern, commencing with the pixel in the bottom right-hand corner, and progressing from right-to-left and left-to-right as we work our way through the chain (Fig.9a).

Y = Rows 0 to 11												
11	133	134	135	136	137	138	139	140	141	142	143	144
10	132	131	130	129	128	127	126	125	124	123	122	121
9	109	110	111	112	113	114	115	116	117	118	119	120
8	108	107	106	105	104	103	102	101	100	99	98	97
7	85	86	87	88	89	90	91	92	93	94	95	96
6	84	83	82	81	80	79	78	77	76	75	74	73
5	61	62	63	64	65	66	67	68	69	70	71	72
4	60	59	58	57	56	55	54	53	52	51	50	49
3	37	38	39	40	41	42	43	44	45	46	47	48
2	36	35	34	33	32	31	30	29	28	27	26	25
1	13	14	15	16	17	18	19	20	21	22	23	24
0	12	11	10	9	8	7	6	5	4	3	2	1
	0	1	2	3	4	5	6	7	8	9	10	11
	X = Columns 0 to 11											

Fig.8. The ordering (numbering) of the NeoPixels in the array.

```
void LightOneAfterAnother (uint32_t thisColor)
{
   for (int iNeo = 1; iNeo < NUM_NEOS; iNeo++)
   {
      Neos.setPixelColor(iNeo, thisColor);
      Neos.show();
      delay(TestCycleTime);
   }
}
```

The main program calls this function over and over again, first setting the colour to red, then green, then blue. If you wish, you can download the full program to peruse and ponder (file **CB-Aug20-01.txt** – available on the August 2020 page of the *PE* website). Also, for your delectation and delight, I created a video showing this in action (**https://bit.ly/3dIpr3G**).

OK, this is where things start to get interesting again. Remember that the way I want to visualise the array – and the way I want to treat it in my programs – is as 12 rows numbered from 0 at the bottom to 11 at the top, and as 12 columns numbered from 0 on the left to 11 on the right. In the future, we want to be able to say things (programmatically speaking) like 'light the pixel at column 4 in row 2 with the colour red.'

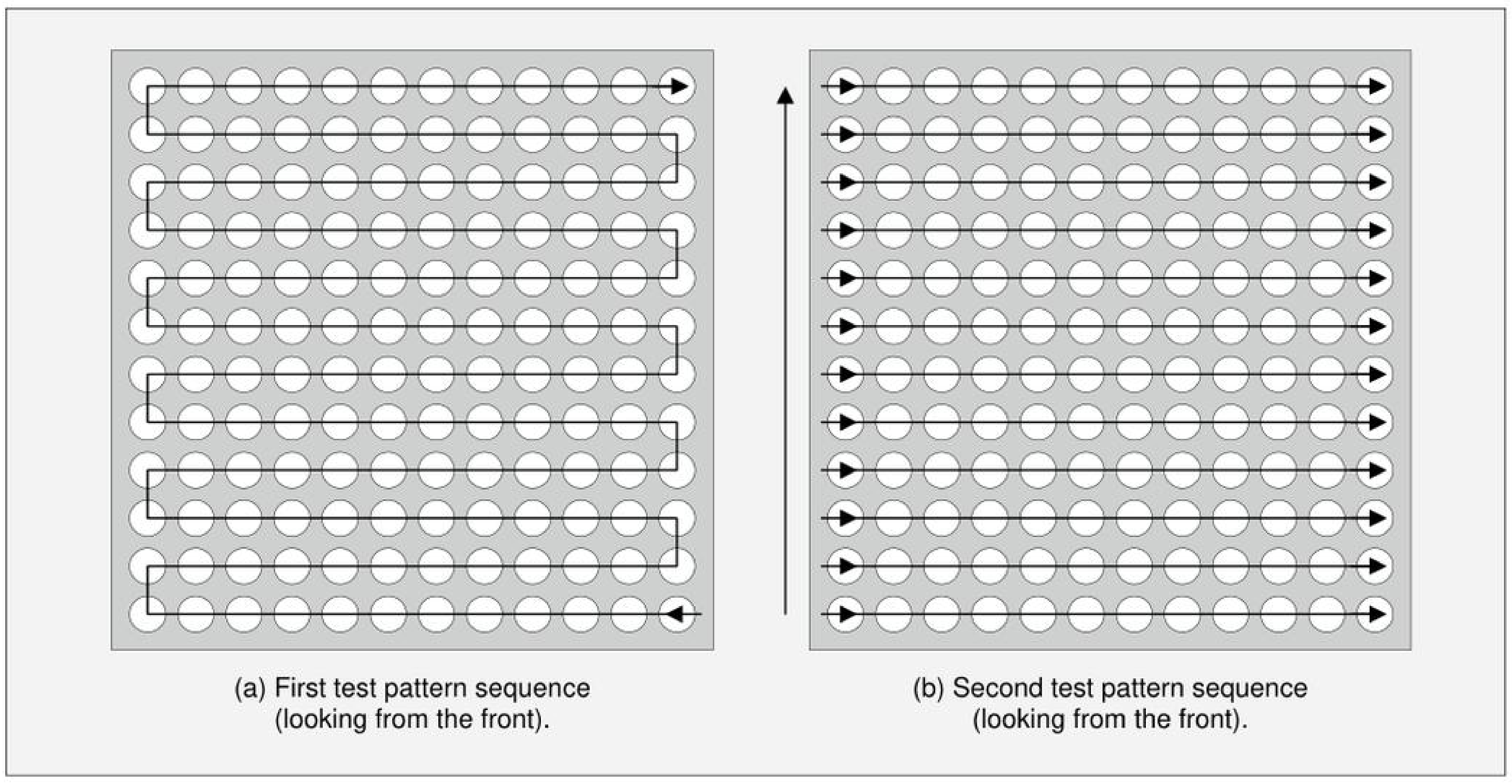
What we want for our second test is to start with row 0 and light the pixels in sequence from column 0 to 11, then to repeat for row 1 and work our way up to row 11. The resulting raster scan should look like the illustration in Fig.9b.

Depending on one's background, you may prefer to think of the column-row combos as X-Y coordinates. In order to achieve this, I modified my main function, which now appears as follows:

```
void LightOneAfterAnother (uint32_t thisColor)
{
   int iNeo;

   for (int yInd = 0; yInd < NUM_ROWS; yInd++)
   {
      for (int xInd = 0; xInd < NUM_COLS; xInd++)
      {
         iNeo = GetNeoNum(xInd, yInd);
         Neos.setPixelColor(iNeo, thisColor);
         Neos.show();
         delay(TestCycleTime);
      }
   }
}
```

As we see, we have an outer loop that works its way up the rows (the Y values), and an inner loop that works its way across the columns (the X values). The interesting part is where we


(a) First test pattern sequence (looking from the front).

(b) Second test pattern sequence (looking from the front).

Fig.9. The results from the first and second tests.

call the `GetNeoNum()` function, passing it the (X,Y) values and – hopefully – receiving the number of the corresponding NeoPixel in the chain.

So, what sort of algorithm could we use to implement the `GetNeoNum()` function? I think that it would be a good exercise for you to cogitate and ruminate on this before reading further. Maybe sketch something out with pencil and paper.

I don't know about you, but this sort of thing doesn't come naturally to me. I'm sure professional programmers could whip it out without thinking, but I'm more of a visual problem solver, so I started off by sketching the number array depicted in Fig.8.

After mulling this over for a while, I decided that if I'm on row Y, my starting point is to say that I have `(Y * 12)` pixels. The next step is to determine if I'm on an odd or even row. If I'm on an even row, I need to add `(12 – X)` pixels; by comparison, if I'm on an odd row, I need to add `(X + 1)` pixels.

The way I determine whether I'm on an odd or even row is to use the `%` (modulo) operator, which returns the integer remainder from an integer division. So, if I divide row `Y` by `% 2` and the result is `0`, we're on an even row; if the result is `1`, we're on an odd row. The code for this function is as follows:

```
int GetNeoNum (int xInd, int yInd)
{
   int iNeo;

   iNeo = yInd * NUM_COLS;

   if ( (yInd % 2) == 0)
   {   // Even row
      iNeo = iNeo + (12 – xInd);
   }
   else
   {   // Odd row
      iNeo = iNeo + (xInd + 1);
   }

   return iNeo;
}
```

Let's try this out. Suppose we want to light the pixel located at column 4 in row 2; that is, `(X,Y) = (4,2)`. First, we multiply the row by the number of pixels, so `(Y * 12) = (2 * 12) = 24`. Next, we divide the row by `% 2` to determine that it's even, in which case we need to add `(12 – X) = (12 – 4) = 8`. So the number of the pixel in the chain that corresponds to (X,Y) coordinates of (4,2) is 24 + 8 = 32. Try this out for yourself using a few sample (X,Y) values and checking the results using Fig.8.

Once again, if you wish, you can download the full program to peruse and ponder (file **CB-Aug20-02.txt** – available on the August 2020 page of the *PE* website). And, once again, I created a short video showing all of this in action (**https://bit.ly/3cFcfLM**).

So, now we're really ready to rock and roll. What shall we do first? I have a few ideas, which I will discuss and demonstrate in my next column. Until then, as always, I welcome your comments, questions, and suggestions.

Cool bean Max Maxfield (Hawaiian shirt, on the right) is emperor of all he surveys at **CliveMaxfield.com** – the go-to site for the latest and greatest in technological geekdom.

Comments or questions? Email Max at: **max@CliveMaxfield.com**

Max's Cool Beans cunning coding tips and tricks

Crusty bits

I have a retired friend, who calls himself 'Crusty', and who is teaching himself to program in C. A few weeks ago he emailed me with a problem. He'd created a program for his Arduino Uno with a `for()` loop that looked something like the following:

```
for (i = 0; i <= 10, i++)
{
   // Do some stuff
}
```

Everything worked as expected with the loop executing 11 times. This wasn't the issue to which I alluded. The problem arose when Crusty modified his code to look something like the following:

```
for (i = 10; i >= 0, i--)
{
   // Do some stuff
}
```

Crusty's issue is that this loop never ended. Instead, it kept on executing over and over again. Any professional programmer will immediately guess the cause. With my decades of painfully gleaned experience, it was obvious to me too, but poor old Crusty simply couldn't figure it out. Can you?

You're not my type!

When people are first introduced to the C programming language, one of the first data types they meet is the `int`, which stands for integer; for example:

```
int MyInt;
```

Variables declared with this type, like `MyInt` in this example, can store positive and negative whole numbers (I know that, by definition, there aren't any negative whole numbers, but you know what I mean). For example, –7, 0, and 42 are all valid `int` values. When we see a number like 42, by convention we assume it represents a positive value without having to explicitly write +42. Similarly, when we see a variable declared using the `int` data type, we assume we're talking about a 'signed' integer that can represent both positive and negative values. We could also make this explicit using the following:

```
signed int MyInt;
```

Of course, this leads us to the fact that we can also declare an integer variable as being unsigned, which means it has no sign and can represent only positive values; for example:

```
unsigned int MyUint;
```

Unlike numbers written with a pencil on paper, which can be as big as we want, limited only by the staying power of our pencil and the endurance of our hand, the size of numbers stored in a computer is limited by the amount of memory associated with the data type we are using to represent them.

Say what?

Now, this is where things start to get a little tricky because – believe it or not – the C standard doesn't explicitly define the size of an `int`. All it says is that an `int` should be a minimum of two bytes (16 bits). In the case of an Arduino Uno, the size of an `int` is indeed two bytes; in other computers it can by four bytes or more.

A 2-byte (16-bit) field can be used to represent 2^{16} = 65,526 different patterns of 0s and 1s. In the case of a `signed int`, these patterns can be used to represent negative and positive values in the range –32,768 to +32,767 (note that we also have to represent 0, so 32,768 + 32,767 + 1 [to represent 0] equals 65,536). By comparison, in the case of an `unsigned int`, these patterns can be used to represent only positive values in the range 0 to 65,535 (once again, we have to represent 0, so 65,535 + 1 [to represent 0] equals 65,536).

All is revealed!

Returning to Crusty's problem (and remembering he only sent me the code for his `for()` loop), it was obvious to me that he'd declared the variable `i` he was using to control his loop as an unsigned data type. I assumed an `unsigned int`, and this indeed turned out to be the case.

Since Crusty was thinking that his loop control would only ever be positive (ie, `>= 0`), he'd fallen into the trap of thinking 'bigger is better,' opting to use an `unsigned int` because it could hold larger positive values, even though he never actually planned on using anything bigger than 10.

Now, let's perform a little thought experiment. We know that an `unsigned int` on an Arduino Uno can be used to represent positive values in the range 0 to 65,565. Suppose we were to load such a variable with 0 and then keep on incrementing (adding one to) it until it contains its maximum value of 65,565. What do you think will happen if we try to increment it one more time? In fact, since the result will exceed this data type's capacity, it will overflow and return to containing 0.

Contra wise, the opposite happens in the other direction. That is, if our `unsigned int` contains 0 and we attempt to subtract 1 from this value, the result can't be –1 because – by definition – our `unsigned int` can contain only positive values. Instead, 0 – 1 will result in 65,565. Although this may not seem particularly intuitive, it actually makes perfect sense once you understand how these values are stored, represented, and manipulated inside the computer, but that's a discussion for another day.

I explained all this to Crusty. I also asked him to add a `Serial.begin(9600);` statement at the beginning of his `setup()` function, and to modify his `for()` loop, as shown below:

```
for (i = 10; i >= 0, i--)
{
   Serial.print(“i = “);
   Serial.println(i);
   // Do some stuff
}
```

When Crusty uploaded this new sketch and launched the Serial Monitor window, the count sequence displayed was as predicted: '…3, 2, 1, 0, 65,535, 65,534…' Of course, this explains why Crusty's loop never terminates, because `i` is always `>= 0`.

A can of worms

Actually, Crusty's question has opened up a can of worms – interesting worms, but worms nonetheless – because we also have signed and unsigned versions of other data types like `short` and `long`.

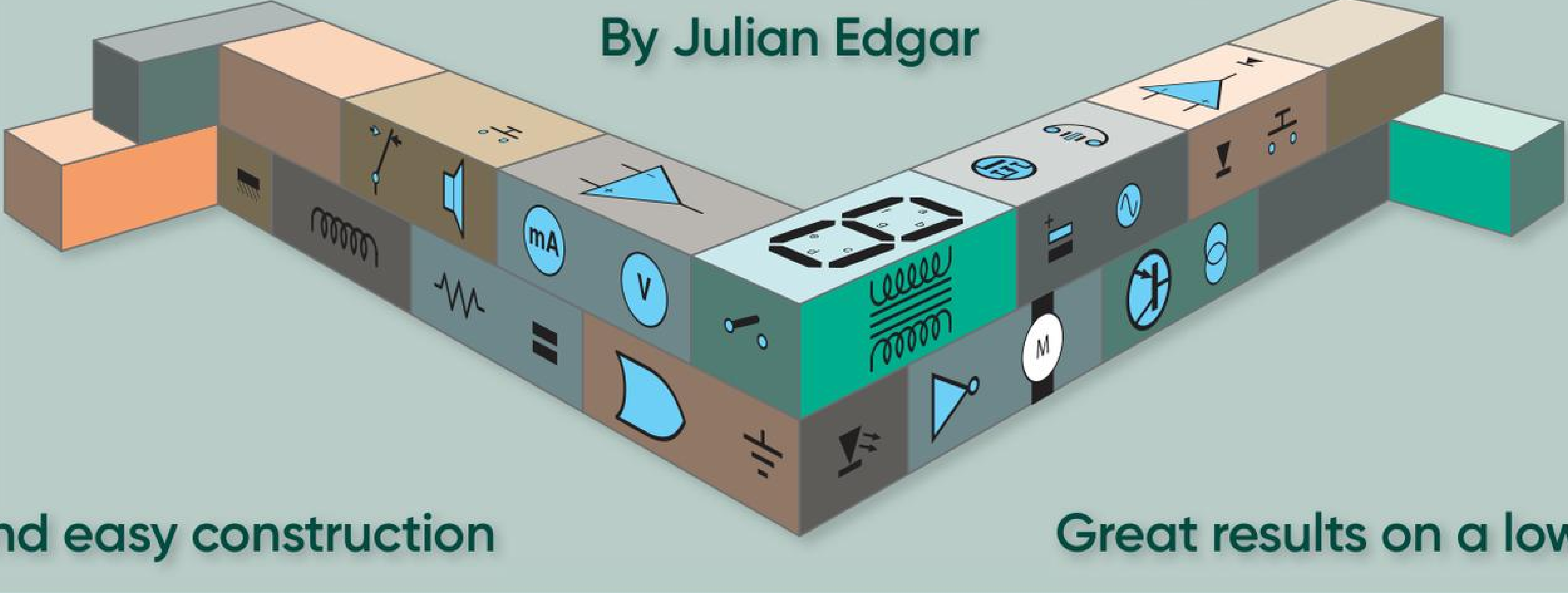
We also have the `char` data type, whose signed or unsigned status is not actually specified by the C standard, which means it can vary from computer to computer or – more correctly – compiler to compiler because the computer just does what it's told. (In the case of the Arduino Uno and its compiler, the `char` is one byte in size and behaves like a signed 8-bit integer.)

And then there's the `byte` data type, which doesn't actually exist in standard C, but which the Arduino's creators decided to throw into the mix for giggles and grins. I'm sure you will be delighted to discover that we'll take a deeper dive into all of this in my next *Tips and Tricks* column.

CRICKLEWOOD ELECTRONICS
Established 1981
Your best bet since MAPLIN
Chock-a-Block with Stock
Visit: www.cricklewoodelectronics.com
Or phone our friendly knowledgeable staff on 020 8452 0161
Components • Audio • Video • Connectors • Cables
Arduino • Test Equipment etc, etc
Visit our Shop, Call or Buy online at:
www.cricklewoodelectronics.com
020 8452 0161
Visit our shop at:
40-42 Cricklewood Broadway
London NW2 3ET

Electronic Building Blocks

By Julian Edgar



Quick and easy construction

Great results on a low budget

Battery capacity tester

If you use batteries in any devices, here is a tool you really must have on your workbench. It's a battery capacity tester, a brilliant piece of test gear that ticks all the boxes.

Cheap? Yes, at round £6 delivered, available from Banggood (ID: 1557935). Versatile? Yes, it can work with batteries having any voltage from 1-15V and any chemistry. You can also set the test discharge current (just use different-value external resistors) and you can also set the endpoint voltage.

Before we look at the device in detail, what use can be made of it? Well, have you ever wondered if it's worth buying expensive brand-name batteries rather than the so-much-cheaper supermarket own-brand variety? Just buy a few examples of each and then test their respective capacities with this module.

Are you doubtful if all rechargeable 18650 cells meet their stated capacity? Buy one that has big claims and test it.

And, just the other day I designed a circuit that drew 100mA from a 9V (PP3) battery. I felt that was probably a bit high for the size of battery – but exactly how long would it last at this current draw? Hard to estimate isn't it – especially when in this application, the battery output can fall as low as 5V and the circuit will still operate. But using the battery capacity tester, we can test a typical PP3 battery at this current draw, setting the endpoint to 5V. Remember also, that battery capacity is dependent on current draw. That is, as discharge increases, capacity decreases (Peukert's law – see Wikipedia). So, having the ability to test a battery under the actual discharge rate that will be experienced when the circuit is in operation is very useful. Finally, as the tester also records time, we can specify what the battery life would be in this application, expressed in minutes of continuous use.

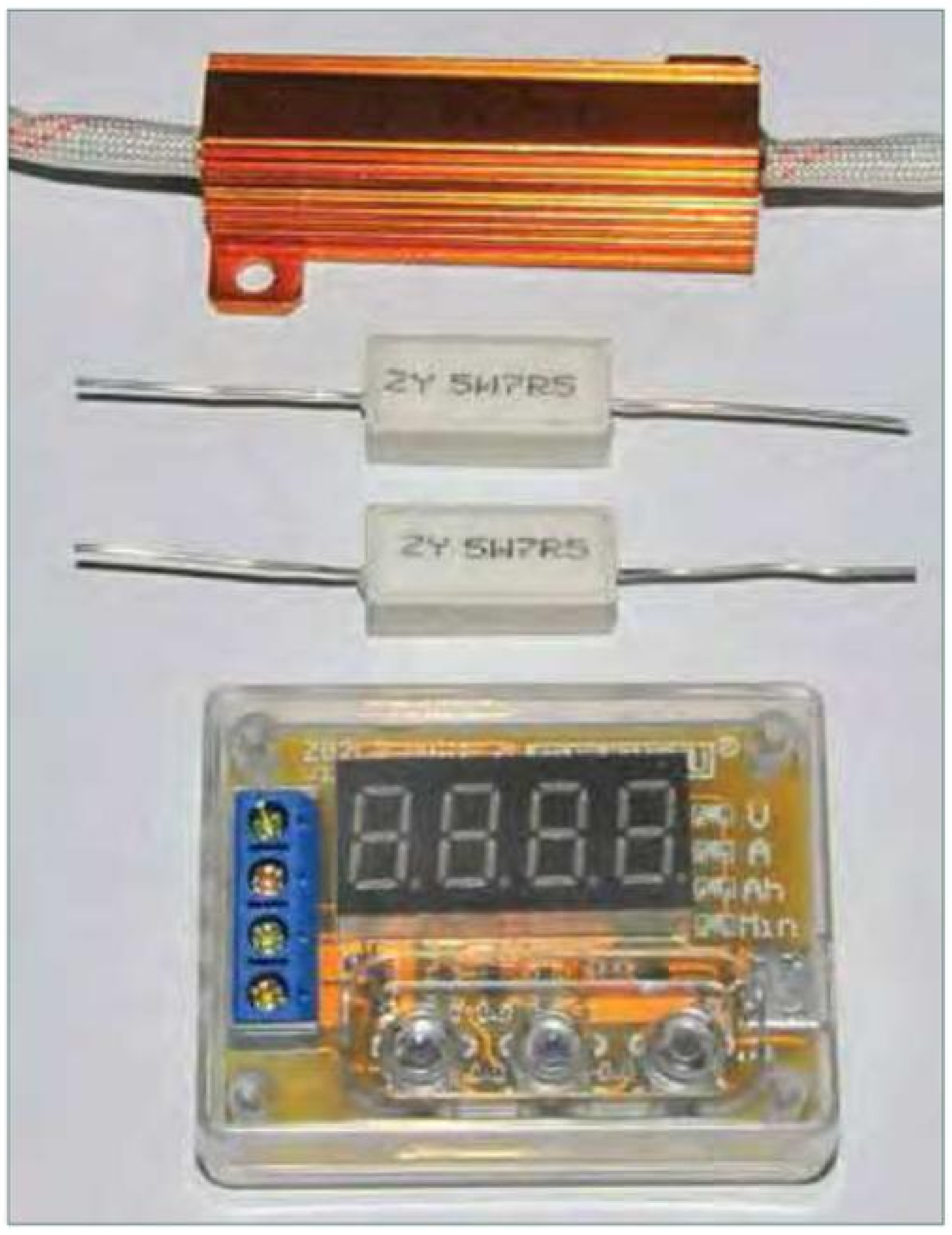


Fig.1. The battery capacity tester (bottom) is supplied with three high-power resistors to act as loads. The tester calculates the battery's amp-hour capacity, and during the test shows real-time battery voltage and current draw. Any battery from 1-15V can be tested at loads of up to 3A.

The module

The unit is a very compact 54 × 41 × 17mm. It comes in a neat clear plastic box with a micro USB socket at one end (no cable supplied) and a 4-connector terminal strip at the other end.

The terminal strip is labelled IN (+ and –) and RL (+ and –). IN is for the connections to the battery under test, and the RL connections are for an external resistor that forms the load. (The connections for the resistor are not polarised, unless you are using an electronic load.)

The device is powered at a nominal 5V via the micro USB socket. Ensure that the power supply won't turn off after a while (eg, from a laptop or PC going into sleep mode) as the module must remain powered throughout the whole test.

Three pushbuttons are provided on the front. These are labelled (–), (+) and OK.

In use

First, determine what load current you wish to use in the test – this will determine the specs of the resistor. Note that current is limited to a maximum of 3A. Ohm's law can be used to determine the required resistor value: resistance (Ω) = voltage (V) divided by current (A). For example, if you wish to discharge a 3V battery at 100mA (0.1A), you would use a resistor with a nominal value of 30Ω. Also keep in mind that the resistor dissipates power. The resistor voltage (V) multiplied by its current (A) gives the power (W), so in this example we are dissipating only 3 × 0.1 = 0.3W – a 30Ω, 1W wirewound resistor will be fine.

The module is supplied with three resistors (two 7.5Ω, 5W, one 6Ω, 50W). Note that at anywhere near their rated power dissipation, these resistors will get very hot. It's best to de-rate them substantially and if you're not able to do that, place them on a china plate or similar so they cannot burn anything!

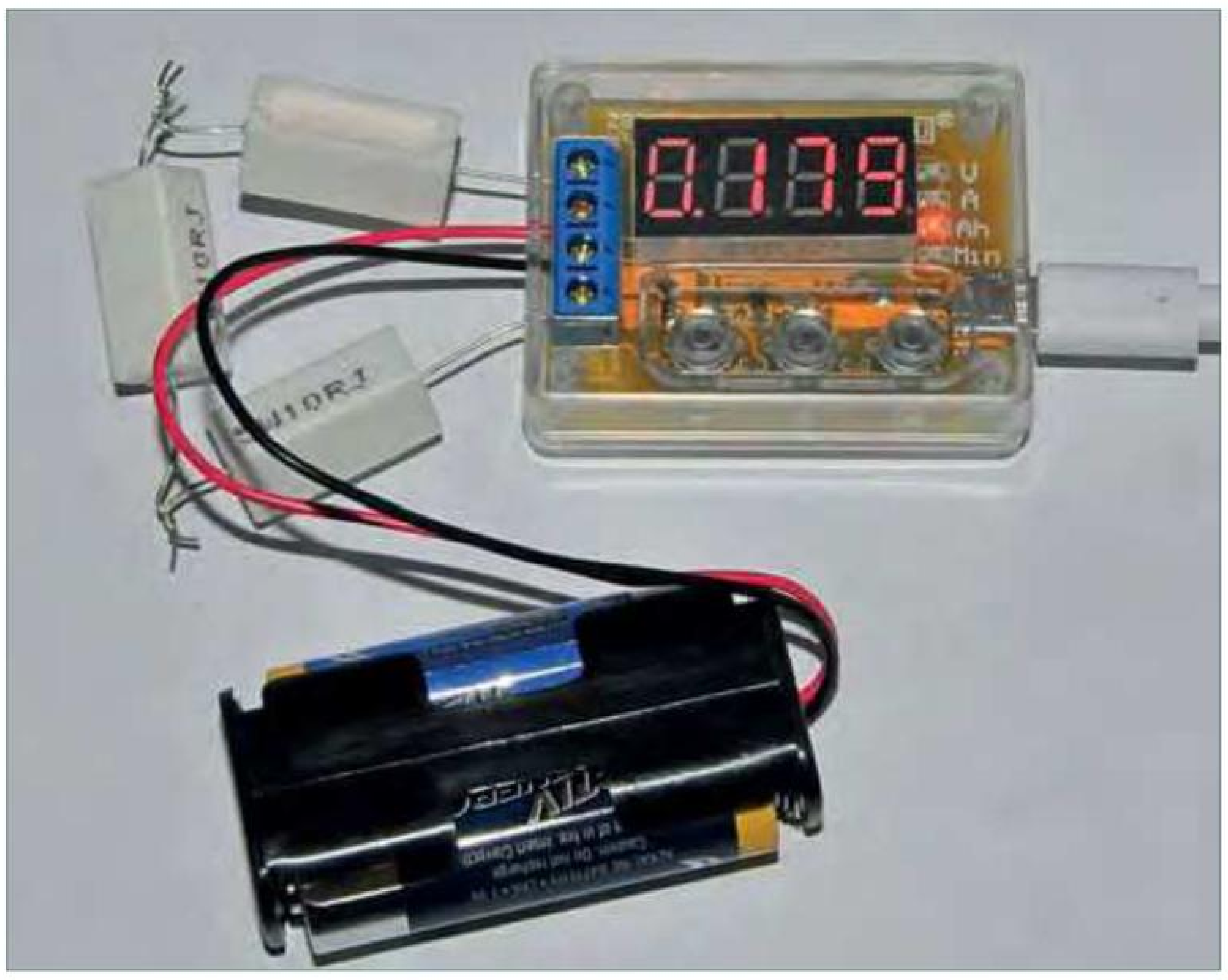


Fig.2. Testing two AA cells. Three 10Ω resistors are being used to give a nominal 100mA load. Towards the end of the test, the display shows 0.179Ah. Power is supplied via the micro-USB cable at right.

Fig.3. Is it worth buying expensive brand-name batteries? This tester can easily find out – I was surprised at its results!

In many uses, you will need resistor values other than these. However, high-power resistors are now available very cheaply online. Incidentally, there's no requirement to use a resistor as the load; for example, you could also use a filament bulb or an electric motor.

After you have connected the battery and load, power-up the tester. The voltage of the battery is then displayed on the LED to two decimal places. By pressing the (–) button, we can then set the endpoint voltage, to one decimal place. (The tester can also automatically set the endpoint voltage, based on the starting voltage of the fully charged battery and presumably using some internal look-up values that guess at battery chemistry. However, it seems to me that it's best if the endpoint voltage is set manually, taking into account the battery type and proposed use. This auto endpoint function can be disabled.) Once the endpoint voltage is set, press OK.

Display

The battery test then starts, and the LED display cycles through the following values:

- The set endpoint voltage (with an 'E' leading figure for 'endpoint')
- Battery voltage
- Current draw
- The amp-hours that have been drawn
- The time (minutes) that the test has been operating

Each of these is indicated by a flashing LED that lights next to an appropriate notation on the board (eg, V, A, Ah).

There are some additional functional settings (eg disabling the display from cycling through the different values) but note that none of these further settings allow calibration of the current or voltage measurements. However, in use, I found the voltage and current readings quite accurate.

The tester can also display error codes. These are:

Err1	Battery voltage is higher than 15V
Err2	Battery voltage is lower than the endpoint voltage
Err3	Load discharge current is too high (presumably this is determined by the battery voltage immediately sagging below the set endpoint)
Err4	Current over 3.1A
Err5	Current sampling or output transistor defective (eg, through reverse-polarity connection)

Note that the Maximum test values are 999.9Ah and 9999 minutes (just under 7 days!).

Example testing

I first tested some brand-new, low-cost supermarket brand AA alkaline cells – two in series, as they are often used. I set the current draw to a nominal 100mA via a 30Ω resistor and the endpoint to 2.8V (ie, 1.4V per cell) – and sat back. I must admit that I found the process quite fascinating (perhaps I am easily entertained), as I watched the battery voltage falling towards the endpoint voltage. Exactly 129 minutes later, the test automatically finished when battery voltage dropped below 2.8V. (Ending of the test is indicated by the LED display flashing. The data can then be cycled through by pressing the (+) or (–) buttons. Don't press 'OK' or you lose the data as the test restarts.)

And the capacity of the new two-cell battery at 100mA current draw? As you'd expect with the above figures, just over 0.2Ah. (Note though that the actual current draw of the resistor of course varies with battery voltage, so it's not just a case of taking into account the time and nominal current draw – the tester is more accurate than that.)

Salvage those cells!

Hmm, so what about some salvaged AA cells? (I often collect batteries that others have thrown away. Typically, about half the batteries in 'recycle' containers are still quite useable, especially for items in use only occasionally.)

This time I started with two 'brand name' cells having a series voltage of 2.9V (ie, 1.45V per cell). If I were to use these, I'd put them in a device where they could be nearly completely discharged before the device ceased to operate. I therefore dropped the endpoint voltage to 2.6V (ie, 1.3V per cell). Test current was as before – a nominal 100mA. And the results? Nearly one-and-a-half hours of use and a measured capacity of 0.13Ah. Not bad for salvaged batteries!

Summary

This is an excellent product – especially considering its versatility and very low cost. True, it is limited to 15V and 3A, but that still makes it suitable for most batteries in general use. If you don't have a selection of high-current resistors on hand, buy them at the same time as you buy the tester, and then you'll have a set-up that will be very effective.

DIRECT BOOK SERVICE

Teach-In 2017

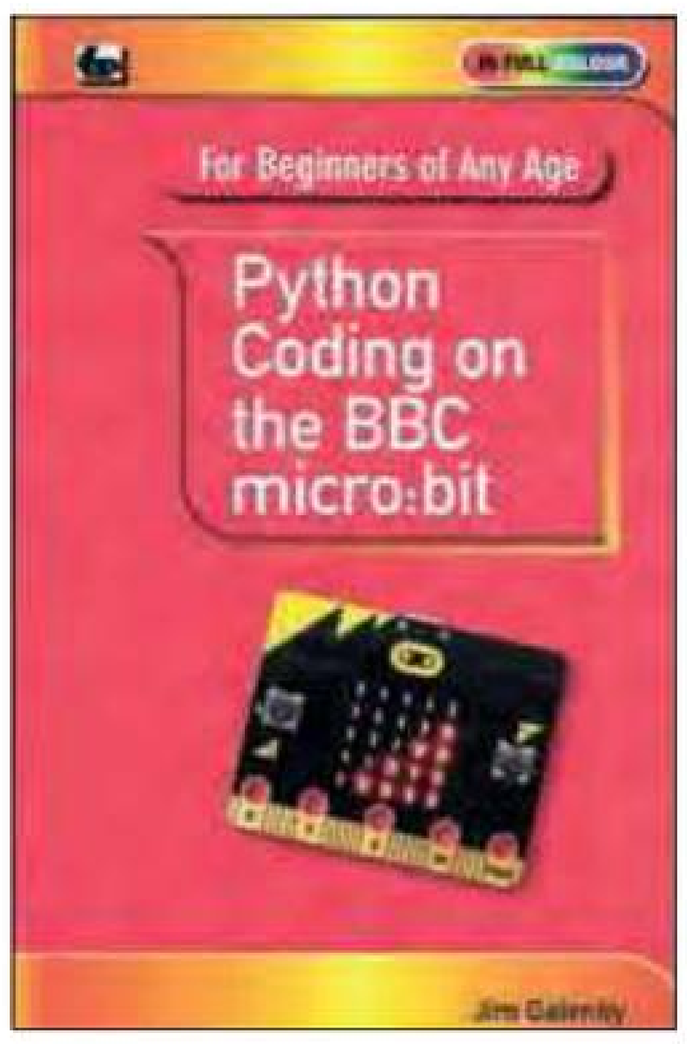
Introducing the BBC micro:bit

The books listed here have been selected by the *Practical Electronics* editorial staff as being of special interest to everyone involved in electronics and computing. They are supplied by mail order direct to your door.

FOR A FULL DESCRIPTION OF THESE BOOKS AND CD-ROMS SEE THE SHOP ON OUR WEBSITE

www.electronpublishing.com

All prices include UK postage



PYTHON CODING ON THE BBC MICRO:BIT

Jim Gatenby

Python is the leading programming language, easy to learn and widely used by professional programmers. This book uses MicroPython, a version of Python adapted for the BBC Micro:bit.

Among the many topics covered are: main features of the BBC micro:bit including a simulation in a web browser screen; various levels of programming languages; Mu Editor for writing, saving and retrieving programs, with sample programs and practice exercises; REPL, an interactive program for quickly testing lines of code; scrolling messages, creating and animating images on the micro:bit's LEDs playing and creating music, sounds and synthesized speech; using the on-board accelerometer to detect movement of the micro:bit on three axes; glossary of computing terms.

This book is written using plain English, avoids technical jargon wherever possible and covers many of the coding instructions and methods which are common to most programming languages. It should be helpful to beginners of any age, whether planning a career in computing or writing code as an enjoyable hobby.

118 Pages **Order code PYTH MBIT** £7.99

GETTING STARTED WITH THE BBC MICRO:BIT

Mike Tooley

Not just an educational resource for teaching youngsters coding, the BBC micro:bit is a tiny low cost, low-profile ARM-based single-board computer. The board measures 43mm × 52mm but despite its diminutive footprint it has all the features of a fully fledged microcontroller together with a simple LED matrix display, two buttons, an accelerometer and a magnetometer.

Mike Tooley's book will show you how the micro:bit can be used in a wide range of applications from simple domestic gadgets to more complex control systems such as those used for lighting, central heating and security applications. Using Microsoft Code Blocks, the book provides a progressive introduction to coding as well as interfacing with sensors and transducers.

Each chapter concludes with a simple practical project that puts into practice what the reader has learned. The featured projects include an electronic direction finder, frost alarm, reaction tester, battery checker, thermostatic controller and a passive infrared (PIR) security alarm.

No previous coding experience is assumed, making this book ideal for complete beginners as well as those with some previous knowledge. Self-test questions are provided at the end of each chapter, together with answers at the end of the book. So whatever your starting point, this book will take you further along the road to developing and coding your own real-world applications.

108 Pages **Order code BBC MBIT** £7.99

THEORY AND REFERENCE

PRACTICAL ELECTRONICS HANDBOOK – 6th Ed
Ian Sinclair
440 pages **Order code NE21** £33.99

STARTING ELECTRONICS – 4th Ed
Keith Brindley
296 pages **Order code ELSEV100** £18.99

ELECTRONIC CIRCUITS – FUNDAMENTALS & APPLICATIONS – Updated version
Mike Tooley
400 pages **Order code TF43** £32.99

FUNDAMENTAL ELECTRICAL AND ELECTRONIC PRINCIPLES – 3rd Ed
C.R. Robertson
368 pages **Order code TF47** £21.99

A BEGINNER'S GUIDE TO TTL DIGITAL ICs
Robert Penfold
142 pages **OUT OF PRINT BP332** £5.45

UNDERSTANDING ELECTRONIC CONTROL SYSTEMS
Owen Bishop
228 pages **Order code NE35** £36.99

MICROPROCESSORS

INTERFACING PIC MICROCONTROLLERS – 2nd Ed
Martin Bates
298 pages **Order code NE48** £34.99

PROGRAMMING 16-BIT PIC MICROCONTROLLERS IN C – LEARNING TO FLY THE PIC24
Lucio Di Jasio (Application Segments Manager, Microchip, USA)
496 pages + CD-ROM **Order code NE45** £38.00

INTRODUCTION TO MICROPROCESSORS AND MICROCONTROLLERS – 2nd Ed
John Crisp
222 pages **Order code NE31** £29.99

THE PIC MICROCONTROLLER YOUR PERSONAL INTRODUCTORY COURSE – 3rd Ed
John Morton
270 pages **Order code NE36** £25.00

PIC IN PRACTICE – 2nd Ed
David W. Smith
308 pages **Order code NE39** £24.99

MICROCONTROLLER COOKBOOK
Mike James
240 pages **Order code NE26** £36.99

BOOK ORDERING DETAILS

All prices include UK postage.

For postage, add £3 per book to Europe, £4 for rest of the world per book.

CD-ROM prices include VAT and/or postage to anywhere in the world.

Send a cheque, (£ sterling only) made payable to: **Practical Electronics** or credit card details (Visa or Mastercard) to:

Electron Publishing Limited,
113 Lynwood Drive, Wimborne, Dorset BH21 1UU

Books are normally sent within seven days of receipt of order.

Please check price (see latest issue of Practical Electronics or website) before ordering from old lists.

For a full description of these books please see the shop on our website.
Tel: 01202 880299 – Email: shop@electronpublishing.com

Order from our online shop at: www.electronpublishing.com

ARDUINO

ARDUINO FOR DUMMIES

John Nussey

Arduino is no ordinary circuit board. Whether you're an artist, a designer, a programmer, or a hobbyist, Arduino lets you learn about and play with electronics. You'll discover how to build a variety of circuits that can sense or control real-world objects, prototype your own product, and even create interactive artwork. This handy guide is exactly what you need to build your own Arduino project – what you make is up to you!

- **Learn by doing** – start building circuits and programming your Arduino with a few easy examples – right away!
- **Easy does it** – work through Arduino sketches line by line, and learn how they work and how to write your own.
- **Solder on!** – don't know a soldering iron from a curling iron? No problem! You'll learn the basics and be prototyping in no time.
- **Kitted out** – discover new and interesting hardware to turn your Arduino into anything from a mobile phone to a Geiger counter.
- **Become an Arduino savant** – find out about functions, arrays, libraries, shields and other tools that let you take your Arduino project to the next level
- **Get social** – teach your Arduino to communicate with software running on a computer to link the physical world with the virtual world

438 Pages **Order code ARDDUM01** **£19.99**

EXPLORING ARDUINO

Jeremy Blum

Arduino can take you anywhere. This book is the roadmap.

Exploring Arduino shows how to use the world's most popular microcontroller to create cool, practical, artistic and educational projects. Through lessons in electrical engineering, programming and human-computer interaction, this book walks you through specific, increasingly complex projects, all the while providing best practices that you can apply to your own projects once you've mastered these. You'll acquire valuable skills – and have a whole lot of fun.

- **Explore** the features of commonly used Arduino boards
- **Use** Arduino to control simple tasks or complex electronics
- **Learn** principles of system design, programming and electrical engineering
- **Discover** code snippets, best practices and system schematics you can apply to your original projects
- **Master** skills you can use for engineering endeavours in other fields and with different platforms

357 Pages **Order code EXPARD01** **£26.99**

COMPUTING AND ROBOTICS

NEWNES INTERFACING COMPANION
Tony Fischer-Cripps
295 pages **Order code NE38** **£41.00**

HOW TO BUILD A COMPUTER MADE EASY
Robert Penfold
120 pages **Order code BP707** **£8.49**

EASY PC CASE MODDING
Robert Penfold
192 pages + CD-ROM **Order code BP542** **£8.99**

FREE DOWNLOADS TO PEP-UP AND PROTECT YOUR PC
Robert Penfold
128 pages **Order code BP722** **£7.99**

WINDOWS XP EXPLAINED
N. Kantaris and P.R.M. Oliver
264 pages **Order code BP514** **£7.99**

THE INTERNET – TWEAKS, TIPS AND TRICKS
Robert Penfold
128 pages **Order code BP721** **£7.99**

eBAY – TWEAKS, TIPS AND TRICKS
Robert Penfold
128 pages **Order code BP716** **£7.50**

AN INTRODUCTION TO eBAY FOR THE OLDER GENERATION
Cherry Nixon
120 pages **Order code BP709** **£8.49**

HOW TO FIX YOUR PC PROBLEMS
Robert Penfold
128 pages **Order code BP705** **£8.49**

AN INTRODUCTION TO WINDOWS VISTA
P.R.M. Oliver and N. Kantarris
120 pages **Order code BP703** **£8.49**

WINDOWS 8.1 EXPLAINED
180 Pages **Order code BP747** **£10.99**

AN INTRODUCTION TO THE NEXUS 7
118 Pages **Order code BP744** **£8.99**

COMPUTING FOR THE OLDER GENERATION
Jim Gatenby
308 pages **Order code BP601** **£8.99**

ANDROIDS, ROBOTS AND ANIMATRONS
Second Edition – John Iovine
224 pages **Order code MGH1** **£16.99**

ROBOT BUILDERS COOKBOOK
Owen Bishop
366 pages **Order code NE46** **£26.00**

INTRODUCING ROBOTICS WITH LEGO MINDSTORMS
Robert Penfold
288 pages + **Order code BP901** **£14.99**

MORE ADVANCED ROBOTICS WITH LEGO MINDSTORMS
Robert Penfold
298 pages **Order code BP902** **£14.99**

HOW TO FIX YOUR PC PROBLEMS
Robert Penfold
128 pages **Order code BP705** **£8.49**

WINDOWS 7 – TWEAKS, TIPS AND TRICKS
Andrew Edney
120 pages **Order code BP708** **£8.49**

GETTING STARTED IN COMPUTING FOR THE OLDER GENERATION
Jim Gatenby
120 pages **Order code BP704** **£8.49**

WINDOWS 8.1 EXPLAINED
Noel Kantaris
180 Pages **Order code BP747** **£10.99**

COMPUTING WITH A LAPTOP FOR THE OLDER GENERATION
Robert Penfold
120 pages **Order code BP702** **£8.49**

AN INTRODUCTION TO EXCEL SPREADSHEETS
Jim Gatenby
18 pages **Order code BP701** **£8.49**

KINDLE FIRE HDX EXPLAINED
118 Pages **Order code BP743** **£8.99**

BOOK ORDER FORM

Full name: ....

Address: ....

....

.... Post code: .... Telephone No: ....

Email: ....

☐ I enclose cheque/PO payable to Practical Electronics for £ ....

☐ Please charge my card £ .... Card expiry date..

Card Number ....

Valid From Date .... Card Security Code .... (The last three digits on or just below the signature strip)

Please send book order codes: ....

....

....

Please continue on separate sheet of paper if necessary

THE BASIC SOLDERING GUIDE

80
The Basic Soldering Guide
Learn to solder successfully!
Alan Winstanley
ANTEX

LEARN TO SOLDER SUCCESSFULLY!

ALAN WINSTANLEY

The No.1 resource for learning all the basic aspects of electronics soldering by hand.

With more than 80 high quality colour photographs, this book explains the correct choice of soldering irons, solder, fluxes and tools. The techniques of how to solder and desolder electronic components are then explained in a clear, friendly and non-technical fashion so you'll be soldering successfully in next to no time! The book also includes sections on reflow soldering and desoldering techniques, potential hazards, useful resources and a very useful troubleshooting guide.

Also ideal for those approaching electronics from other industries, the *Basic Soldering Guide Handbook* is the best resource of its type, and thanks to its excellent colour photography and crystal clear text, the art of soldering can now be learned by everyone!

86 Pages **Order code AW1** **£9.99**

VISIT OUR WEBSITE FOR MORE BOOKS AND FAST, EASY ONLINE ORDERING: www.electronpublishing.com

Practical Electronics PCB SERVICE

Summer Sale

PROJECT	CODE	PRICE
AUGUST 2020		
Micromite LCD BackPack V3	07106191	£7.95
Steering Wheel Audio Button to Infrared Adaptor	05105191	£7.95
JULY 2020		
AM/FM/CW Scanning HF/VHF RF Signal Generator	04106191	£11.95
Speech Synthesiser with the Raspberry Pi Zero	01106191	£5.95
PE Mini-organ PCB	AO-0720-01	£14.95
PE Mini-organ selected parts	AO-0720-02	£8.95
High-current Solid-state 12V Battery Isolator – control	05106191	£6.95
High-current Solid-state 12V Battery Isolator FET (2oz)	05106192	£9.95
JUNE 2020		
Arduino breakout board – 3.5-inch LCD Display	24111181	£6.95
Six-input Audio Selector main board	01110191	10.95
Six-input Audio Selector switch panel board	01110192	
MAY 2020		
Ultra low distortion Preamplifier Input Selector	01111112	[illegible]1[illegible]25
Ultra low distortion Preamplifier pushbutton Input Selector	01111113	
Universal Regulator	18103[illegible]	[illegible]
433MHz Wireless Data Repeater	1[illegible]91	[illegible]8.50
Bridge mode Adaptor for Amplifier	[illegible]105[illegible]	[illegible].95
iCEstick VGA Terminal	[illegible]0319	[illegible]4.95
Analogue noise with tilt control	AO[illegible]520-01	[illegible]7.95
Audio Spectrum Analyser	PM-0520-01	[illegible]8.95
APRIL 2020		
Flip-dot Display black coil board	19111181	[illegible]95
Flip-dot Display black pix[illegible]	19111182	
Flip-dot Display black [illegible]me	191111[illegible]	
Flip-dot Display green [illegible]	191[illegible]84	
MARCH 2020		
Diode Curve Plotter	0[illegible]218[illegible]	£10.95
Steam Train Whistle / Diesel Horn Sound Generator	09[illegible]181	£8.50
Universal Passive Crossover (one off)	[illegible]0320	£12.50
Crossover component set for Wavecor speaker (one off)	WAVXO (see website)	
FEBRUARY 2020		
Motion-Sensing 12V Power Switch	05102191	£5.95
USB Keyboard / Mouse Adaptor	24311181	£8.50
DSP Active Crossover (ADC)	01106191	£29.95
DSP Active Crossover (DAC) ×2	01106192	
DSP Active Crossover (CPU)	01106193	
DSP Active Crossover (Power/routing)	01106194	
DSP Active Crossover (Front panel)	01106195	
DSP Active Crossover (LCD)	01106196	
JANUARY 2020		
Isolated Serial Link	24107181	£8.50
DECEMBER 2019		
Extremely Sensitive Magnetometer	04101011	£16.75
Four-channel High-current DC Fan and Pump Controller	05108181	£8.75
Useless Box	08111181	£11.50
NOVEMBER 2019		
Tinnitus & Insomnia Killer (Jaycar case – see text)	01110181	£8.75
Tinnitus & Insomnia Killer (Altronics case – see text)	01110182	£8.75
OCTOBER 2019		
Programmable GPS-synced Frequency Reference	04107181	£11.50
Digital Command Control Programmer for Decoders	09107181	£8.75
Opto-isolated Mains Relay (main board)	10107181	£11.50
Opto-isolated Mains Relay (2 × terminal extension board)	10107182	

PROJECT	CODE	PRICE
AUGUST 2019		
Brainwave Monitor	25108181	£12.90
Super Digital Sound Effects Module	01107181	£5.60
Watchdog Alarm	03107181	£8.00
PE Theremin (three boards: pitch, volume, VCA)	PETX0819	£19.50
PE Theremin component pack ([illegible]p 56, August 2019)	PETY0819	£15.00
JULY 2019		
Full-wave 10A Universal [illegible]or[illegible]ed[illegible]ontroller	10102181	£12.90
Recurring Event R[illegible]der	19107181	£8.00
Temp[illegible]re S[illegible]ch	05105181	£10.45
JU[illegible]E 2019		
Ardu[illegible]C Meter	04106181	£8.00
USB Flexitime	19106181	£10.45
MAY 2019		
2× 12V Battery Balancer	14106181	£5.60
Deluxe Frequency Switc[illegible]	05104181	£10.45
USB Port Protect[illegible]	07105181	£5.60
APRIL 2019		
Heater C[illegible]olle[illegible]	10104181	£14.00
MAR[illegible]20[illegible]		
[illegible]0-LED[illegible]graph Main Board	04101181	£11.25
[illegible]+[illegible]ocessing Board	04101182	£8.60
[illegible]BRUARY 2019		
1.[illegible]W Induction Motor Speed Controller	10105122	£35.00
NOVEMBER 2018		
Super-7 AM Radio Receiver	06111171	£27.50
OCTOBER 2018		
6GHz+ Touchscreen Frequency Counter	04110171	£12.88
Two 230VAC MainsTimers	10108161 10108162	£12.88
SEPTEMBER 2018		
3-Way Active Crossover	01108171	£22.60
Ultra-low-voltage Mini LED Flasher	16110161	£5.60
AUGUST 2018		
Universal Temperature Alarm	03105161	£7.05
Power Supply For Battery-Operated Valve Radios	18108171 18108172 18108173 18108174	£27.50
JULY 2018		
Touchscreen Appliance Energy Meter – Part 1	04116061	£17.75
Automotive Sensor Modifier	05111161	£12.88
JUNE 2018		
High Performance 10-Octave Stereo Graphic Equaliser	01105171	£15.30
MAY 2018		
High Performance RF Prescaler	04112162	£10.45
Micromite BackPack V2	07104171	£10.45
Microbridge	24104171	£5.60
APRIL 2018		
Spring Reverberation Unit	01104171	£15.30
DDS Sig Gen Lid	Black	£8.05
DDS Sig Gen Lid	Blue	£7.05
DDS Sig Gen Lid	Clear	£8.05

PCBs for most recent *PE*/*EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.

Back issues of articles are available – see Back Issues page for details.

Double-sided | plated-through holes | solder mask

PROJECT	CODE	PRICE
MARCH 2018		
Stationmaster Main Board	09103171	£17.75
+ Controller Board	09103172	
SC200 Amplifier Module Power Supply	01109111	£16.45
FEBRUARY 2018		
GPS Synchronised Analogue Clock Driver	04202171	£12.88
High Power DC Motor Speed Controller Part 2		
+ Control Board	11112161	£12.88
+ Power Board	11112162	£15.30
JANUARY 2018		
High Power DC Motor Speed Controller Part 1	11112161	£12.88
Build the SC200 Amplifier Module	01108161	£12.88
DECEMBER 2017		
Precision Voltage and Current Reference Part 2	04110161	£15.35
NOVEMBER 2017		
50A Battery Charger Controller	11111161	£12.88
Micropower LED Flasher (45 × 47mm)	16109161	£8.00
(36 × 13mm)	16109162	£5.60
Phono Input Converter	01111161	£8.00
SEPTEMBER 2017		
Compact 8 Digit Frequency Meter	04105161	£12.88
AUGUST 2017		
Micromite Based Touch screen Boat Computer GPS	07[illegible]02122	£1[illegible]5
Fridge/Freezer Alarm	03[illegible]161	£8.[illegible]
JULY 2017		
Micromite Based Super Clock	07102122	£10.45
Brownout Protector for Induction Motors	10107161	£12.90
JUNE 2017		
Ultrasonic Garage Parking Assistant	07102122	£10.45
Hotel Safe Alarm	03106161	£8.00
100dB Stereo LED Audio Lev[illegible] VU [illegible]r	01104161	£17.75
MAY 2017		
The Micromite LCD BackPa[illegible]	07102122	£11.25
Precision 230V/115V 50/60Hz [illegible]ntable Driver	04104161	£19.35
APRIL 2017		
Microwave Leakage Detector	04103161	£8.00
Arduino Multifunctional 24 bit Measuring Shield	04116011	£17.75
+ RF Head Board	04116012	
Battery Pack Cell Balancer	11111151	£9.00
MARCH 2017		
Speech Timer for Contests & Debates	19111151	£16.42
FEBRUARY 2017		
Solar MPPT Charger/Lighting Controller	16101161	£17.75
Turntable LED Strobe	04101161	£7.60
JANUARY 2017		
High performance Stereo Valve Preamplifier	01101161	£17.75
High Visibility 6 Digit LED Clock	19110151	£16.42
DECEMBER 2016		
Universal Loudspeaker Protector	01110151	£12.88
9 Channel Infrared Remote Control	15108151	£16.42
Revised USB Charger	18107152	£5.36

Go online for

PROJECT	CODE	PRICE
NOVEMBER 2016		
Fingerprint Access Controller Main Board	03109151	£12.88
Fingerprint Access Controller Switch Board	03108152	
OCTOBER 2016		
Arduino Based USB Electrocardiogram	07108151	£9.79
100W Switchmode/Linear Bench Supply Part 2	18104141	£20.83
SEPTEMBER 2016		
LED Party Strobe	16101141	£9.80
Speedo Corrector	05109131	£12.00
AUGUST 2016		
Low cost Resistance Reference	04108151	£5.36
USB Power Monitor	04109121	£12.00
JULY 2016		
Driveway Monitor Detector Unit	15105151	£11.80
Driveway Monitor Receiver Unit	15105152	£7.50
USB Charging Points	18107151	£5.00

For the many pre-2016 PCBs that we stock please see the PE website: www.electronpublishing.com

PE/EPE PCB SERVICE

Order Code	Project	Quantity	Price

Name ....

Address ....

....

Tel ....

Email ....

I enclose payment of £ .... (cheque/PO in **£ sterling only**)

payable to **Practical Electronics**

Card No ....

Valid From ....Expiry Date ....

Card Security No ....

You can also order PCBs by phone, email or via the shop on our website: www.electronpublishing.com

No need to cut your issue – a copy of this form is just as good!

All prices include VAT and UK p&p. Add £4 per *project* for post to Europe; £5 per *project* outside Europe.

Orders and payment should be sent to:

Practical Electronics, Electron Publishing Ltd
113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU
Tel 01202 880299 Email: shop@electronpublishing.com
On-line Shop: www.epemag.com

Cheques should be made payable to 'Practical Electronics' **(Payment in £ sterling only)**.

NOTE: Most boards are in stock and sent within seven days of receipt of order, please allow up to 28 days delivery if we need to restock.

ETI BUNDLE (1) Teach-In 3, 4 and 5 – all on CD-ROM – only £18.95

ELECTRONICS TEACH-IN 3 – CD-ROM

Mike & Richard Tooley

The three sections of the *Teach-In 3* CD-ROM cover a huge range of subjects that will interest everyone involved in electronics – from newcomers to the hobby and students to experienced constructors and professionals.

The first section (80 pages) is dedicated to *Circuit Surgery*, *EPE/PE's* regular clinic dealing with readers' queries on circuit design problems – from voltage regulation to using SPICE circuit simulation software.

The second section – *Practically Speaking* – covers hands-on aspects of electronics construction. Again, a whole range of subjects, from soldering to avoiding problems with static electricity and identifying components is covered. Finally, our collection of *Ingenuity Unlimited* circuits provides over 40 circuit designs submitted by readers.

The CD-ROM also contains the complete *Electronics Teach-In 1* book, which provides a broad-based introduction to electronics in PDF form, plus interactive quizzes to test your knowledge and TINA circuit simulation software (a limited version – plus a specially written TINA Tutorial).

The *Teach-In 1* series covers everything from electric current through to microprocessors and microcontrollers, and each part includes demonstration circuits to build on breadboards or to simulate on your PC.

ELECTRONICS TEACH-IN 4 – CD-ROM
A BROAD-BASED INTRODUCTION TO ELECTRONICS

Mike & Richard Tooley

The *Teach-In 4* CD-ROM covers three of the most important electronics units that are currently studied in many schools and colleges. These include, Edexcel BTEC level 2 awards and the electronics units of the Diploma in Engineering, Level 2.

The CD-ROM also contains the full *Modern Electronics Manual*, worth £29.95. **The Manual** contains over 800 pages of electronics theory, projects, data, assembly instructions and web links.

A package of exceptional value that will appeal to anyone interested in learning about electronics – hobbyists, students or professionals.

ELECTRONICS TEACH-IN 5 – CD-ROM
JUMP START

Mike & Richard Tooley

15 design and build circuit projects for newcomers or those following courses in school and colleges.

The projects are: ■ Moisture Detector ■ Quiz Machine ■ Battery Voltage Checker ■ Solar-Powered Charger ■ Versatile Theft Alarm ■ Spooky Circuits ■ Frost Alarm ■ Mini Christmas Lights ■ iPod Speaker ■ Logic Probe ■ DC Motor Controller ■ Egg Timer ■ Signal Injector Probe ■ Simple Radio Receiver ■ Temperature Alarm.

PLUS

PIC'n' Mix – starting out with the popular range of PIC microcontrollers and *Practically Speaking* – tips and techniques for project construction.

The CD-ROM also contains:

■ Complete *Teach-In 2* book, a practical introduction to PIC microprocessors

■ MikroElektronika, Microchip and L-Tek PoScope software.

Three Teach-ins for the great price of £18.95

PLUS you also get the contents of the free CD-ROM from each issue...

... so that's another TWO Teach-Ins and The Full Modern Electronics Manual!

What a Bargain!!

ETI BUNDLE (2) Teach-In 6, 7 and 8 – all on CD-ROM – only £18.95

ELECTRONICS TEACH-IN 6 – CD-ROM
A COMPREHENSIVE GUIDE TO RASPBERRY Pi

Mike & Richard Tooley

Teach-In 6 contains an exciting series of articles that provides a complete introduction to the Raspberry Pi, the low cost computer that has taken the education and computing world by storm.

This latest book in our *Teach-In* series will appeal to electronics enthusiasts and computer buffs who want to get to grips with the Raspberry Pi.

Teach-In 6 is for anyone searching for ideas to use their Pi, or who has an idea for a project but doesn't know how to turn it into reality. This book will prove invaluable for anyone fascinated by the revolutionary Pi. It covers:

■ Pi programming
■ Pi hardware
■ Pi communications
■ Pi Projects
■ Pi Class
■ Python Quickstart
■ Pi World
■ ...and much more!

The *Teach-In 6* CD-ROM also contains all the necessary software for the series, so that readers and circuit designers can get started quickly and easily with the projects and ideas covered.

ELECTRONICS TEACH-IN 7 – CD-ROM
DISCRETE LINEAR CIRCUIT DESIGN

Mike & Richard Tooley

Teach-In 7 is a complete introduction to the design of analogue electronic circuits. It is ideal for everyone interested in electronics as a hobby and for those studying technology at schools and colleges. The CD-ROM also contains all the circuit software for the course, plus demo CAD software for use with the *Teach-In* series.

■ Discrete Linear Circuit Design
■ Understand linear circuit design
■ Learn with 'TINA' – modern CAD software
■ Design simple, but elegant circuits
■ Five projects to build:
  i) Pre-amp
  ii) Headphone Amp
  iii) Tone Control
  iv) VU-meter
  v) High Performance Audio Power Amp.

PLUS

Audio Out – an analogue expert's take on specialist circuits

Practically Speaking – the techniques of project building.

ELECTRONICS TEACH-IN 8 – CD-ROM
INTRODUCING THE ARDUINO

Mike & Richard Tooley

Hardware: learn about components and circuits
Programming: powerful integrated development system
Microcontrollers: understand control operations
Communications: connect to PCs and other Arduinos.

Teach-In 8 is an exciting series designed for electronics enthusiasts who want to get to grips with the inexpensive, popular Arduino microcontroller, as well as coding enthusiasts who want to explore hardware and interfacing. It will provide a one-stop source of ideas and practical information.

The Arduino offers a truly effective platform for developing a huge variety of projects; from operating a set of Christmas tree lights to remotely controlling a robotic vehicle through wireless or the Internet.

Teach-In 8 is based around a series of practical projects with plenty of information for customisation.

This book also includes *PIC n' Mix*: 'PICs and the PICkit 3 – A Beginners guide' by Mike O'Keefe and *Circuit Surgery* by Ian Bell – 'State Machines part 1 and 2'.

The CD-ROM includes the files for:

■ *Teach-In 8*
■ Microchip MPLAB IDE XC8 8-bit compiler
■ PICkit 3 User Guide
■ Lab-Nation Smartscope software.

ORDER YOUR BUNDLE TODAY!
JUST CALL 01202 880299 – OR VISIT www.electronpublishing.com

CLASSIFIED ADVERTISING

Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value. The rate for semi-display space is £10 (+VAT) per centimetre high, with a minimum height of 2·5cm. All semi-display adverts have a width of 5.5cm. The prepaid rate for classified adverts is 40p (+VAT) per word (minimum 12 words).

Cheques are made payable to 'Practical Electronics'. **VAT must be added**. Advertisements with remittance should be sent to: Practical Electronics, 113 Lynwood Drive, Wimborne, Dorset, BH21 1UU.

Tel 01202 880299 Email: stewart.kearn@wimborne.co.uk

For rates and further information on display and classified advertising please contact our Advertisement Manager, Stewart Kearn – see below.

Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.

We have been the leading monthly magazine in this market for the last twenty-seven years.

BOWOOD ELECTRONICS LTD
Suppliers of Electronic Components
www.bowood-electronics.co.uk
Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield, Derbyshire S40 2QR. Sales: 01246 200 222
Send large letter stamp for Catalogue

MISCELLANEOUS

VALVES AND ALLIED COMPONENTS IN STOCK. Phone for free list. Valves, books and magazines wanted. Geoff Davies (Radio), tel. 01788 574774.

PIC DEVELOPMENT KITS, DTMF kits and modules, CTCSS Encoder and Decoder/Display kits.
Visit www.cstech.co.uk

Electrical Industries Charity (EIC)
We help people working in the electrical, electronics and energy community as well as their family members and retirees.
We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.
www.electricalcharity.org
electrical industries charity

COAST ELECTRONICS
BREAKOUTS-COMPONENTS-
CONTRACT DESIGN-3D PRINTER PARTS-
MUSICAL-MICROCONTROLLERS
WWW.COASTELECTRONICS.CO.UK

Andrew Kenny – Qualified Patent Agent
EPO UKIPO USPTO
Circuits Electric Machinery Mechatronics
Web: www.akennypatentm.com
Email: Enquiries@akennypatentm.com
Tel: 0789 606 9725

ADVERTISING INDEX

CRICKLEWOOD ELECTRONICS 63
ESR ELECTRONIC COMPONENTS 36
HAMMOND ELECTRONICS Ltd 8
JPG ELECTRONICS 72
MICROCHIP . . . . Cover (ii)
PEAK ELECTRONIC DESIGN . . . . Cover (iv)
POLABS D.O.O. 51
SILICON CHIP 5
STEWART OF READING 9
TAG-CONNECT 36

Advertisement offices
Electron Publishing Ltd
113 Lynwood Drive
Merley
Wimborne,
Dorset BH21 1UU

Tel 01202 880299
Fax 01202 843233
Email pe@electronpublishing.com
Web www.electronpublishing.com

For editorial contact details see page 7.

Next Month – in the September issue

Radio Head Unit Dimmer

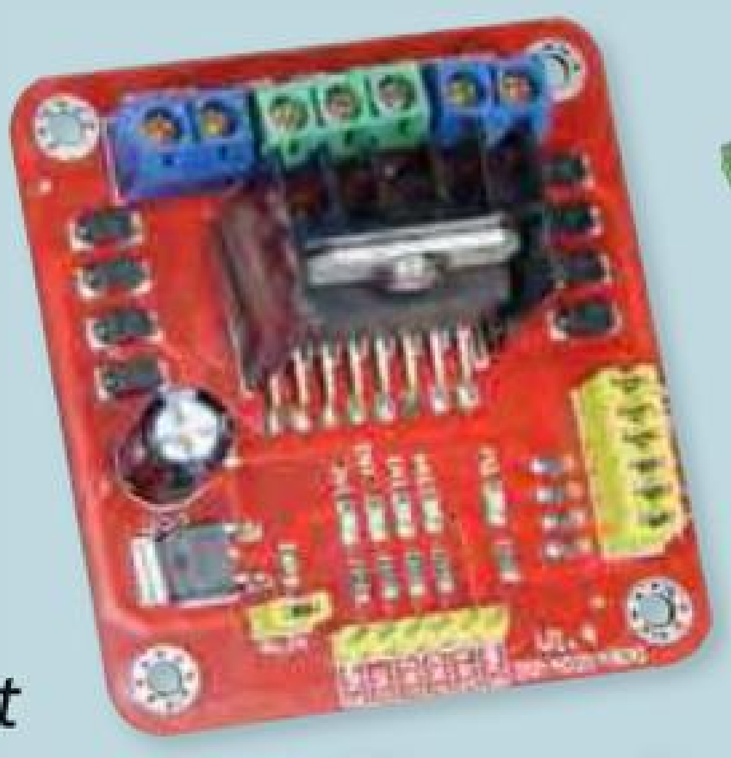
Very few aftermarket car radio 'head units' offer a dimming function, which makes driving in the country at night downright hazardous. This simple device fixes that, adjusting the display and backlighting brightness as you dim your instrument lights. It can also be used as a basic voltage interceptor for various automotive sensors.

Ultrabrite LED Pushbike Light

This tiny (22 × 12mm) circuit board is a high-efficiency LED driver that delivers a constant 1A or 2.2A. You can use it with a 12V white LED array to make a (very!) bright bicycle light, a torch or another light source powered from a lithium-ion or LiPo battery pack.

The Micromite Explore-28

The 28-pin Micromite is a low-cost, powerful microcontroller which allows you to create advanced devices with minimal effort. Now the Explore-28 will make your life even easier. It's a small plug-in module with the same powerful PIC plus a USB socket for comms and programming.



Three Stepper Motor Drivers

Want to build your own 3D printer or CNC machine? You'll need stepper motors, and here are three of the most common stepper motor driver modules and guidance on how to use them.

PLUS!

All your favourite regular columns from *Audio Out, Cool Beans and Circuit Surgery,* to *Electronic Building Blocks*, *PIC n' Mix* and *Net Work*.

On sale 6 August 2020

Content may be subject to change

WELCOME TO JPG ELECTRONICS
Selling Electronics in Chesterfield for 29 Years

Open Monday to Friday 9am to 5:30pm
And Saturday 9:30am to 5pm

- Aerials, Satellite Dishes & LCD Brackets
- Audio Adaptors, Connectors & Leads
- BT, Broadband, Network & USB Leads
- Computer Memory, Hard Drives & Parts
- DJ Equipment, Lighting & Supplies
- Extensive Electronic Components
  - *ICs, Project Boxes, Relays & Resistors*
- Raspberry Pi & Arduino Products
- Replacement Laptop Power Supplies
- Batteries, Fuses, Glue, Tools & Lots more...

T: 01246 211 202
E: sales@jpgelectronics.com
JPG Electronics, Shaw's Row,
Old Road, Chesterfield, S40 2RB
W: www.jpgelectronics.com

JPG Electronics

Retail & Trade Welcome • Free Parking • Google St View Tour: S40 2RB

NEW subscriptions hotline!

Practical Electronics

We have changed the way we sell and renew subscriptions. We now use 'Select Publisher Services' for all print subscriptions – to start a new subscription or renew an existing one you have three choices:

1. Call our NEW print subscription hotline: 01202 087631, or email: pesubs@selectps.com
2. Visit our shop at: www.electronpublishing.com
3. Send a cheque (payable to: 'Practical Electronics') with your details to:
   Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 01202 880299 or visit: www.electronpublishing.com

Published on approximately the first Thursday of each month by Electron Publishing Limited, 1 Buckingham Road, Brighton, East Sussex BN1 3RA. Printed in England by Acorn Web Offset Ltd., Normanton WF6 1TW. Distributed by Seymour, 86 Newman St., London W1T 3EX. Subscriptions UK: £26.99 (6 months); £49.85 (12 months); £94.99 (2 years). EUROPE: airmail service, £30.99 (6 months); £57.99 (12 months); £109.99 (2 years). REST OF THE WORLD: airmail service, £37.99 (6 months); £70.99 (12 months); £135.99 (2 years). Payments payable to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom. Email: pesubs@selectps.com. PRACTICAL ELECTRONICS is sold subject to the following conditions, namely that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price shown on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter whatsoever.